AD-A116 767 WOODS HOLE OCEANOGRAPHIC INSTITUTION MA F/G 9/2
COMPUTER STORAGE AND RETRIEVAL OF POSITION - DEPENDENT DATA.(U)
JUN 82 R C GROMAN N00014-79-C-0071
UNCLASSIFIED WHOI-82-27 NL

1 of 2
AD A
116 767

UNCLASSIFIED 6/82

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER WHOI-82-27	2. GOVT ACCESSION NO. AD-A116 767
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) COMPUTER STORAGE AND RETRIEVAL OF POSITION - DEPENDENT DATA	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert Carl Groman	8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0071; N00014-82-C-0019;
9. PERFORMING ORGANIZATION NAME AND ADDRESS Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 083-004
11. CONTROLLING OFFICE NAME AND ADDRESS NORDA/National Space Technology Laboratory Bay St. Louis, MS 39529	12. REPORT DATE June 1982
	13. NUMBER OF PAGES 139
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-82-27.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

1. Data retrieval
2. Location-dependent data
3. Geographical retrieval

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See reverse side.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED 6/82

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

WHOI-82-27

COMPUTER STORAGE AND RETRIEVAL OF POSITION - DEPENDENT DATA

by

Robert Carl Groman

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

June 1982

TECHNICAL REPORT

Prepared for the Office of Naval Research under Contracts N00014-79-C-0071; NR 083-004 and N00014-82-C-0019; NR 083-004 and JOI, Inc.

Approved for Distribution: Earl E. Hays

Earl E. Hays, Chairman
Department of Ocean Engineering

WOODS HOLE OCEANOGRAPHIC INSTITUTION
WOODS HOLE, MASSACHUSETTS 02543

PREFACE

This thesis covers the design of a new digital database system to replace the merged (observation and geographic location) record, one file per cruise leg, magnetic tape based system which served the Geology and Geophysics Department from 1974 to 1981. This system (described in WHOI Technical Report 74-68 "The Digital Data Library System: Library Storage and Retrieval of Digital Geophysical Data" by Robert C. Groman) provided a relatively simple procedure for areal searching of our bathymetric, geomagnetic and gravity observations. It grew out of the resources available to the staff in the late 60's - 16-bit mini-computers with paper and magnetic tape drives for shipboard acquisition; and an Institution central computer with limited disk storage capacity. It became increasing cumbersome and expensive to retrieve data from small geographic areas due to tape handling required and the sequential data access.

Realizing that convenient and rapid access to the information, together with ease of updating, were essential to the continuing use and cost justification of the department database, and that its sheer size (greater than two million points) made its use via the older system tedious, Bob Groman started in 1978 to assess the options for a new one. We were also interested in integrating other data types, including our seismic profiling and seafloor samples. He began with a series of interviews with the staff and students; surveyed systems used at other laboratories and in related fields; reviewed published computer databases; and looked at the big changes in computer technology, especially in disk drives. Out of this review emerged the design for the database treated in this thesis, which has now been implemented on one of our D.E.C. VAX-11/780 computers. None of the published databases had the capabilities required and so it was necessary to design one permitting insertion by cruise leg and retrieval by either geographic bounds or cruise leg. Appendix VI gives the current detailed design specifications.

A major portion of the costs of the new database came from our Ocean Industry Program, and along with our other supporting agencies, we provide the member companies dial-up access to the data. Some support came from Office of Naval Research and the Ocean Margin Drilling contracts. Partial support of Groman's studies at Worcester Polytechnic Institute came from the WHOI Education Office.

We are pleased to have this new facility in place, and would like by this report to share it with others. I hope that readers will feel free to contact Bob Groman for further information.

John I. Ewing, Chairman
Department of Geology and Geophysics

28 May 1982

ABSTRACT

A data storage and retrieval scheme has been designed and implemented which provides cost effective and easy access to location-dependent, 'geophysical' data. The system is operational on a Digital Equipment Corporation VAX-11/780 computer. Values of measured and computed geophysical parameters, such as geomagnetic field, water depth and gravity field, are stored in the library system. In addition, information about the data, such as port stops, project name and funding agency are also saved. These data are available to a time sharing computer user, validated to use the software package, through a query language designed to interact with this data library. The data can be searched and retrieved both sequentially and geographically.

COMPUTER STORAGE AND RETRIEVAL OF

POSITION-DEPENDENT DATA

by

Robert Carl Groman

A Thesis

Submitted to the faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

degree of Master of Science

in

Computer Science

Robert Carl Groman

May 1981

APPROVED:

Greg W. Scragg

Dr. Greg W. Scragg, Major Advisor

Stephen R Alpert

Dr. Stephen R. Alpert, Head of Department

FOREWORD

A computer storage and retrieval scheme for location-dependent data has been implemented and is described here. Computer routines were written to store hundreds of thousands of oceanographic values in direct access disk files. Data structures were designed to access these data efficiently by geographical area and by acquisition platform name.

Chapter I provides an introduction to the problem and gives a brief description of the solution. Chapter 2 gives a short description of existing geographic management systems. I also review the objectives and features of data base management systems. Chapter 3 describes the user requirements and the user characteristics for a geophysical library system in more detail. The unique characteristics of geophysical data are further defined. Other design criteria such as hardware constraints and funding support are also covered. Chapter 4 describes the implementation features of the library system including a description of the storage scheme and key algorithms. Chapter 5 provides a summary of the design specifications and a comparison between this new scheme and existing systems.

I would like to acknowledge the help of my advisor for many useful conversations during all phases of this work. Donna Allison, who was able to read my writing, deserves credit for the care she took in typing this thesis. My special thanks go to my wife, Susan, for her patience, support and understanding that were needed to keep both of us going.

The Woods Hole Oceanographic Institution's Ocean Industry Program, the Education Office and the Office of Naval Research provided support for some of this work.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter I

INTRODUCTION

Data management of oceanographic data shares many common features with data management in other scientific and business disciplines. However, there are some unique problems and features of oceanographic data which require special attention. For example, the data, collected by oceangoing ships, are position and time dependent. Because of the uncertainties in the ship's position and measuring instruments, it may be necessary to save different values for the same physical phenomenon. Also, the number of measurements that need to be stored and later retrieved are in the millions.

Geophysical measurements including 'navigation', 'depths', 'geomagnetics' and 'gravity', are used by geologists, geophysicists and other physical scientists to understand the basic physical processes occurring on and below the earth's surface. In order to take advantage of the increasing amount of geophysical measurements collected by oceanographic institutions, government agencies and private research groups, an efficient and flexible approach to data storage and retrieval is needed (Hartman, 1978). The following is a description of a system I designed and implemented to satisfy these storage and retrieval requirements.

Problem Description

The primary attribute of geophysical measurements is their position, given as a pair of latitude and longitude values. These two numbers taken together can be viewed as the unique key for the information about a position on the earth's surface, much in the same way as a person's social security number fully identifies that individual. All geophysical data must be viewed as having this location dependency. The information cannot be acquired, stored, retrieved or analyzed without considering at what geographical location the data were collected. This two dimensional aspect to geophysical data requires that an ordered pair of numbers, the latitude and longitude, be associated with each data value if the values are to constitute useful measurements. It is reasonable to ask whether a general purpose data base management system (Date, 1976) might not be suitable for implementing a computer based data storage and retrieval system for these location-dependent data. However, the nature of the location dependency, the nature of the data parameters, and the way these data are accessed and used make such a solution impractical.

Geophysical data are commonly collected by the oceanographic research community using surface ships or airplanes. Because the acquisition platform is usually moving during data collection, these data are often called 'underway data'. Measurements are collected directly in digital or analog form. These measurements include the 'corrected depth', 'total geomagnetic field', 'total gravity field', 'current velocity', and many more.[1] Information is collected continuously with a repetition rate that depends on the rate of change of the particular attribute. For example, the 'total geomagnetic field' is typically recorded aboard ship every five minutes over deep water (greater than 1000 meters) but every one minute over shallower water since the measured values vary faster in shallower water. Other data parameters have acquisition rates ranging from every ten seconds (e.g. ship's velocity) to every few hours (e.g. bird sightings).

Additional information is derived by further data processing or combining two or more of the measurements. For example, the 'free air anomaly' is obtained by subtracting the 'total gravity field' from a theoretically computed 'regional gravity field'. Also, new measurements may evolve in future times. For example, one new procedure is to add a second

[1] Appendix I provides an explanation of many of the oceanographic terms used here.

magnetometer (an instrument used to measure 'geomagnetic field') towed a fixed distance from the first magnetometer. This arrangement allows a 'gradient geomagnetic value' to be computed and is used to further understand the subbottom sea structure.

As is evident from the above, a data storage scheme for geophysical data must be able to manage a large number of different parameters at each geographical position. In addition, because of the variability in acquisition rates and the needs of the individual researchers, the data storage scheme must allow for missing data parameters.

The data storage scheme is further complicated by the retrieval requirements that researchers place on a geophysical digital data library. Investigators often make requests about the existence of geophysical data as well as requests to retrieve these data within certain geographical boundaries. Since data are collected in chronological order, a request for chronological retrieval is also routine.

Data retrieval by chronological order is used during the analysis of ocean bottom features. Indeed, the original cruise track was probably chosen to simplify this type of post-cruise analysis. A data structure using sequential data storage easily satisfies this type of request. However, area studies based on a synthesis of data from many different cruises require that data be accessible by geographic area. A data structure which allows direct access to the stored values is needed to avoid lengthy, sequential searches for relevant information. An additional requirement is that data retrieval cannot be expensive since certain query requests can return many thousands of data values.

Overview of Solution

The system which I implemented and which is described here is designed to satisfy the needs of the oceanographic research community for a digital storage and retrieval system for location-dependent data. The key features include the following:

1. A data dictionary describes each element stored in the library. This approach allows for easy growth when new data attributes are added.

2. A data storage scheme, using magnetic disk storage and isolating the positional information from the secondary measurements, provides flexible and efficient data storage. Infrequently used data can be removed from on-line storage without jeopardizing the ability to satisfy requests about the data.

3. The data organization facilitates retrieval within geographical boundaries and yet retains a sequential access capability.

4. Data compaction techniques reduce disk storage needs and reduce retrieval costs.

5. An inquiry facility provides the user with a comprehensive set of commands to query the library about the information stored as well as provide interfaces to existing graphics routines.

A crucial consideration in designing a data retrieval system is how the data will be used. In my case, there may be requests for profiles and charts, and summaries of data availability. One can also expect that new applications will introduce new location-dependent parameters. It is the position, the location on the earth's surface at which the parameter was measured, which plays a central role in these data sets. This position-dependency leads to two types of relations: a near neighbors relation and a sequential neighbors relation.

The near neighbor relation refers to the fact that the study of oceanography very often analyzes the relationship among data values within geographically close locations. Depending on the features under investigation, the geographical area can span a few hundred square kilometers ('fracture zone studies') to as much as an entire ocean (Mid-Atlantic Ridge studies). Data obtained by different researchers, over the course of many years, are often studied together in order to piece together an understanding of the basic physical processes taking place on, in, and below the ocean.

The sequential neighbors relation occurs because data parameters are collected, usually aboard a sea-going ship, in sequence, as the ship moves through the water. Often, the change in the values from one position to the next is as informative as the absolute value of the collected data. This leads to a requirement that data be accessible in the same order as they were collected.

These two relationships form the basis of nearly all existing underway geophysical retrieval requests at the Woods Hole Oceanographic Institution. It is not surprising then, that I have designed the data storage scheme to facilitate retrieval along these lines. However, I have not lost sight of the fact that requirements evolve and new data types are added, and I have included features in my design to allow for change.

Chapter II

GENERAL SOLUTION TECHNIQUES

There are always advantages and disadvantages in structuring data in any given way. Since different data have different characteristics, this should affect the data organization (Date, 1976 and Martin, 1975). Furthermore, specific applications for data will often require unique access methods. For example, in the preparation of a contour plot, the contouring task will be simplified if the data are presorted into an evenly spaced grid. However, if data from many different sources are geographically sorted, retrieval from a single source is not enhanced. Because of this variability in data characteristics and data applications, it is not surprising that a number of data management schemes have been produced. In this chapter, four geophysical data management schemes are described in order to provide additional insight into the problem of storing and retrieving location-dependent data. I begin by considering the features of geographical data processing schemes. I then conclude with a discussion of the four geophysical data management systems described previously.

Geographical Data Processing

The broad area of geographical data processing and management is of interest to city planners who want to manage city growth, acquisition and use of census studies, studies of regional land use, as well as the oceanographic applications considered here. The storage schemes used by geographical data processing systems are classified by Nagy and Wagle into two broad categories: data banks and data bases (Nagy and Wagle, 1979). In a data bank, data are separated according to some natural intellectual understanding of the data or according to a subextent (i.e., geographical) basis or both. In general, data banks are simpler to implement than data bases since they are specifically designed to input, output and process data along the lines of the data separation. Data typically are stored sequentially although provision for selective retrieval usually is provided. In a data base implementation, data are stored along with inter-entity relational information. These relations enhance the utility of data and offer other benefits.

Many geographical data processing systems divide the data into convenient manageable geographical areas. The Experimental Cartographic System (ECS), developed in the later part of the 1960's, uses 15 by 15 minute areas (Nagy and Wagle, 1979). The Storage and Access of Network Data on Rivers and Drainage Basins system (STANDARD) divides data into either 15 or 7 1/2 minute areas for processing (Wagle, 1978). More flexibility is provided by the Geodata Analysis and Display System (GADS), which used a relational data base management system (Nagy and Wagle, 1979). In GADS the basic geographical unit's size and shape can be adjusted to each application.

These and other geographical data processing systems do share some common attributes with business data processing systems (Nagy and Wagle, 1979). Common attributes include the presence of many complex data interrelationships, the need for reliable and stable production settings, and a tendency toward interactive systems. However, the two dimensional quality of the data destined for a geographical data processing system sets it apart from other forms of information. This quality deeply influences the input and output operations, algorithm design and especially the data organization.

Geophysical Data Management Systems

This section describes four existing oceanographic geophysical data management schemes. Figure 1 provides a summary comparison of these four schemes. Each system shares two common attributes: they make use of some form of data location table to speed access to stored data; and they rely on sequential access to the stored data.

The Lamont-Doherty Geological Observatory (LDGO) stores 'navigation', 'corrected depth', 'geomagnetics' and 'gravity' measurements on removable magnetic disk packs (Weissel, 1979). Data records are organized by cruise leg, a typical approach. A cruise leg constitutes one continuous acquisition period, by a single ship, usually from one port stop to the next. Data collected during a cruise leg are stored as a single sequential file. A data location table, stored on disk, acts as an index to the various cruises and contains information about the cruise such as project name, start and end dates, and the name of the funding agency. The location table contains the geographical boundaries which encompass each cruise leg. A retrieval by area is handled by searching the location table to determine which cruise leg boundaries overlap the requested retrieval area. By using the location table, a preprocessor program is able to generate an input parameter file for various plotting programs. This procedure has two major benefits:

1) it hides the organization and complexity of the data storage system; and
2) it simplifies running of plotting routines which, because of their flexibility, usually require complex and numerous run-time parameters.

Figure 1

Comparison of Existing Oceanographic Geophysical Data Management Schemes

	Uses a Location Table	Primary Storage Medium	Allows Variable Data format	Data Access Method
LDGO	Yes	Disk	No	Sequential
NGSDC	Yes	Tape[1]	No[2]	Sequential
SIO	Yes	Tape	Yes	Sequential
WHOI	Yes	Tape	No	Sequential

[1] Abbreviated summary of cruise tracks exists on disk

[2] Not at this time although there is provision for this feature.

The storage and retrieval scheme used by the Scripps Insitution of Oceanography (SIO) also uses the cruise leg as the basic organizational unit (Smith, 1979). However, unlike LDGO's scheme, each cruise leg record is allowed to take its own form within the cruise leg file. That is, while each logical record in a cruise leg will have the same format, this format need not be the same format as used in another cruise leg. There exists a file definition scheme which specifies the length and contents of the records of each data file. Obviously, this scheme benefits from being able to accept new data attributes. The SIO scheme further differs from the LDGO scheme in that actual data records reside on magnetic tape rather than disk. However, like LDGO, access to the cruise leg files is facilitated by a location table maintained on disk.

The geophysical library at the Woods Hole Oceanographic Institution (WHOI) uses a tape-based data library approach (Groman, 1974). A disk file, similar to LDGO's location file, stores information about each cruise leg in addition to information about where data are stored. The retrieval scheme searches the location table to see which cruise legs fall into the area of interest and provides the tape names and file numbers for these cruises.

The national repository for marine geology and geophysics data, the National Geophysical and Solar-Terrestrial Data Center (NGSDC), collects and manages depth, geomagnetic, gravity and other data provided by oceanographic, educational, commercial, and other government organizations (Bender, 1979). The storage scheme used at NGSDC stores data on magnetic tape but maintains a disk-based abbreviated summary of cruise tracks. With this data organization they can provide researchers with information about where and how much data exists. However, actual data are not stored on-line and must be retrieved sequentially from magnetic tape.

In all of these schemes, the cruise leg is the basic data storage unit. While this philosophy allows easy and inexpensive storage and retrieval by cruise leg, it does little to facilitate retrieval by geographical boundaries. In each case, once a particular cruise leg is identified as containing information within the specified area, every record in the cruise leg file must be accessed. That is, the basic unit of data retrieval is the whole cruise leg. This can become very inefficient, especially when only a small percentage of points within the 'cruise leg' are actually used.

Underway geophysical measurements possess many unique features. These include:

1. the two dimensional nature of the data - each data point must be associated with a geophysical position;

2. large amounts of data - a cruise leg will collect digital data at 5,000 to 35,000 distinct geographical positions;

3. non-uniform collection methods - acquisition rates may differ for each parameter measured; and,

4. retrieval requests require that data be accessible sequentially by cruise leg and geographically by geographical bounds.

I have been unable to locate any data management system that successfully handles location-dependent data collected over a very large area. The schemes described above attempt to resolve the somewhat incompatible requirements of accessing data both sequentially by cruise legs and randomly by geographic location.

Chapter III

MAJOR LIBRARY DESIGN CONSIDERATIONS

Many factors were considered during the design of this storage and retrieval facility. This section describes six of the more important areas including: user attributes, data attributes, adding new data, modifications and deletions, computer hardware, and retrieval costs. Appendix II outlines the general library design specifications.

User Attributes

In order to help determine the retrieval capabilities that a new geophysical data management scheme should possess, I took a survey of potential users at the Woods Hole Oceanographic Institution. These users were asked to complete a questionnaire and, in some cases, users were interviewed by me in person or by phone.

The results of this survey indicate that easy access to stored geophysical data is the primary goal although easy access does not necessarily mean immediate access to all data. The system should handle queries about the existence of data as well as support the various graphical display options currently available (charts and profiles). No one could answer questions about their query frequency with precision. Answers ranged from once per year to once per week. However, I expect that once a retrieval scheme is implemented which provides researchers with enhanced retrieval and graphical tools, the query frequency will increase.

The discussions with potential users suggest that any new scheme should allow new location-dependent data to be added to the library without extensive changes. This is evidenced by the interest in managing other data types, such as 'core' and 'dredge' data, 'bottom photographs', 'bottom samples' and 'sedimentary analysis' data. These data are now managed using a combination of computer and non-computer techniques and it is reasonable to expect that once these collections outgrow their present management systems they might benefit from data management software similar to that described here.

What Data Should Be Stored

Part of the preliminary system design process which I conducted included a study of the attributes which should be stored in a library of underway geophysical measurements. Figure 2 summarizes the geophysical data types and quantitities currently stored by WHOI's existing tape-based system. This storage scheme requires 162 megabytes of tape storage, although there does exist considerable data redundancy. I estimate that 80 Mbytes of storage would be needed if redundancies were eliminated.

Figure 2

Data Quantities at WHOI

Data Category	Logical Record Length	Number of Cruise Leg Files	Total Number of Records	Total Number of Bytes
Navigation	64	151	83,889	5,368,896
Bathymetry	72	130	801,769	57,727,368
Magnetics	72	117	735,933	52,987,176
Gravity	92	82	499,622	45,965,224

Appendix II summarizes the data attributes which are currently considered to be of interest to geophysicists (NOAA, 1977). Since this list is a superset of the data now available from the WHOI tape library, it is clear that any new scheme would require at least 80 Mbytes for data storage.

Adding New Data

It is difficult to specify how often newly acquired geophysical data will be added to the library. When geophysicists make underway geophysical measurements at sea, new data are generated at an average rate of 1225 "measurements" per day or 10,000 sets of position-dependent records per 'cruise leg'. The number of geophysical-type cruise legs per year at WHOI varies between 0 and 20.

It is a unique feature of geophysical measurements that there may exist non-unique values for measurements. This can happen because of the errors inherent in the navigational and measuring instruments. The library must be able to save all data values, even if it results in conflicting measurements.

Modifications and Deletions

The rate of data record modification (including deletions) is equally difficult to determine. Based on the existing tape-based library there would be less than a 1% modification rate to a 'cruise leg' and a 0% deletion rate. Once data are collected and added to the library, they are never completely removed.

Computer Hardware

Any computer system, as long as it supports large disk storage with a random access capability and a high level programming language, could be a target machine for the library system described here. The system selected was a Digital Equipment Corporation's VAX-11/780 computer because it was available and provided disk storage.

Retrieval Costs

Part of my design time was spent studying plot requests for geophysical data. Figure 3 shows a summary of three such requests. Any new scheme should be able to satisfy this type of request and, it is hoped, at a reduced cost. I realize this reduced cost on my new scheme by increasing the hit ratio. Although storage and access costs are higher on a per record basis for disk versus tape, by reducing the number of accesses needed to fulfill a request, a savings is realized.

Figure 3

Case Study: Data Retrieval and Plot Tape Generation

	Area 1	Area 2	Area 3
No. files read	5	14	16
Hit ratio after SEARCH (percent)[1]	5.8	3.9	8.5
Hit ratio without SEARCH (percent)[2]	0.1	0.4	0.9
Total cost[3]	$20.07	$72.88	$72.22

---

[1] Program SEARCH is first run to determine which cruise legs are likely to lie in the area specified. The hit ratio is defined as the number of useful records decoded (that is, data points which actually fall within the requested area) divided by total number of records read, deblocked and decoded, times 100.

[2] This hit ratio is computed as the number of useful records decoded divided by the total number of logical records in the library, times 100. This figure would apply if no mechanism were used to preselect cruise legs.

[3] Total cost is the cost to retrieve data and create a plot tape used for off-line plotting. These figures are based on 1979 computer rates at Woods Hole Oceanographic Institution for the Sigma 7.

Data which are accessed infrequently can migrate from on-line disk storage to lower cost tape storage. Based on experience with Woods Hole Oceanographic Institution's existing tape based library, I anticipate that less than 50 percent of primary data and 70 percent of secondary data need be on-line at any one time, although some information about each 'cruise leg' must always be available on-line. I estimate that 300 megabytes are needed for on-line disk storage. This figure is based on an analysis of the disk storage requirements. Figure 4 summarizes this analysis.

Disk storage costs, as well as updating and maintenance costs, can be charged to the users of the system with the record keeping and charge-back mechanism which I have built into the retrieval software.

Figure 4

Analysis of Disk Storage Requirements

Assumptions:

1. 151 cruise legs are to be stored.
2. Provide growth for 300 cruise legs.
3. Average cruise leg is 28 days long, collecting 15,008 position records.
4. Average ship speed is 6 knots.
5. Four detail groups will be stored: geophysical detail, bathymetry detail, geomagnetic detail and gravity detail data.
6. Data are collected in every 10 degree square.
7. Infrequently used data can migrate to tape storage.

Data Structure	Disk Space Required (bytes)
Ship Table	412,672
Bounds Tables[1]	31,186,944
Primary Data	1,473,408 per leg
Secondary Data	1,440,768 per leg

Total disk space for all 151 cruise legs: 471,640,000 bytes

By removing infrequently used data from disk storage, the on-line storage needs are satisfied by one 300 megabyte disk drive.

---

1 Each bounds table provides room for 1000 entries. Based on the assumptions above, it takes 10 hours to traverse a degree square. Hence approximately 68 entries are needed per cruise leg. Certain bounds tables will grow faster than others, while other bounds tables will not be needed at all.

application of the data will define whether data as close as 1 mile or 100 miles are considered to be "near". For example, location-dependent data collected for census studies would resolve distances to a different degree than an analysis of global weather patterns (Alvarez and Taylor, 1974; Nagy and Wagle, 1979). For the geophysical data library implemented here, a one-degree by one-degree area (approximately 60 nautical miles by 60 nautical miles at the equator) defines the basic grid size. The data located within a one degree area can then be considered as near.

Obviously, this division into one degree shapes is arbitrary. I could just as well have chosen one degree shapes which center on the intersection of integral latitude and longitude lines. However, my choice reflects the common approach. A more important decision was that of choosing one degree shapes as opposed to larger or smaller geographical areas. By choosing the one degree division I balanced the need for high hit ratios during data retrieval against the maintenance, storage and complexity costs inherent in choosing very small base areas (Wagle, 1978).

The organization I chose to implement the geographical proximity relationship is a data structure called the bounds table. This structure views the earth's surface as 10 degree by 10 degree areas for a total of 648 areas covering the application of the data will define whether data as close as 1 mile or 100 miles are considered to be "near". For example, location-dependent data collected for census studies would resolve distances to a different degree than an analysis of global weather patterns (Alvarez and Taylor, 1974; Nagy and Wagle, 1979). For the geophysical data library implemented here, a one-degree by one-degree area (approximately 60 nautical miles by 60 nautical miles at the equator) defines the basic grid size. The data located within a one degree area can then be considered as near.

Obviously, this division into one degree shapes is arbitrary. I could just as well have chosen one degree shapes which center on the intersection of integral latitude and longitude lines. However, my choice reflects the common approach. A more important decision was that of choosing one degree shapes as opposed to larger or smaller geographical areas. By choosing the one degree division I balanced the need for high hit ratios during data retrieval against the maintenance, storage and complexity costs inherent in choosing very small base areas (Wagle, 1978).

The organization I chose to implement the geographical proximity relationship is a data structure called the bounds table. This structure views the earth's surface as 10 degree by 10 degree areas for a total of 648 areas covering the earth. Each of these areas is further divided into 1 degree by 1 degree areas, or one hundred subareas per area. Access to data by geographical area is accomplished using the following steps:

Step 1. Choose the 10 by 10 degree area based on the desired geographical position.

Step 2. Select the 1 by 1 degree area within the 10 by 10 degree defined above.

In the implementation of the bounds table I use separate random access files for each 10 degree by 10 degree area. Step 1 in the algorithm is a mapping from a latitude and longitude into a file name. The sign of the latitude determines whether the file name contains an 'N' (north for positive) or 'S' (south for negative) in position 4 of the file name. The sign of the longitude determines whether the file name contains an 'E' (east for positive) or 'W' (west for negative) in position 6 of the name. The most significant digit of the latitude and the two most significant digits of the longitude are used in positions 5 and 7 through 8, respectively, to uniquely define the bounds table file names. Figure 5 is a diagram of a bounds table.

Access to data within the one degree squares is accomplished by accessing one of 100 pointer records contained within the file.

Figure 5

Bounds Table

The name of the bounds table uniquely specifies which 10 degree by 10 degree area is being accessed.

Record 1	Free record pointer, creation data, date last updated
Record 2[1]	Pointer for degree 1
Record 101	Pointer for degree 100
Record 102[2]	

[1] These records contain a count of the total number of navigational records in each degree square plus a pointer to the start of individual linked lists beginning after record 101.

[2] The remaining records in the table are part of linked lists and contain records which store the total number of records available for each data type for each degree square and a pointer record into the primary data files each time a cruise leg enters the degree square.

A simple formula, implementing step 2 above, is used to access this information:

linear position within file = absolute (latitude units) times 10 + absolute (longitude units) + 2

Additional facts are maintained in these tables so that user queries about amounts and types of data available can be answered without having to access the original data records. The data structure used within a file is a linked list. It is easy to maintain and update and provides sufficient response times for typical queries. A single bounds table file, containing information about 100 separate degree squares, must be at least 101 records long. (Each record is 48 bytes long.) One additional record is needed each time a cruise leg passes into a degree square. This record contains a pointer to the position data. A separate record is maintained for each data type in order to record how many data points exist in the degree square.

This scheme allows for the fact that there will often be many 10 degree by 10 degree areas containing no data. In such a case, the bounds table for this area need not exist. This reduces the overhead due to sparse data collections.

The second relationship, the cruise leg organization of the data, provided the basis for many early geophysical data storage schemes. While each scheme uses its own way to identify a collection period, either by project name or 'cruise leg', the period invariably describes a continuous interval, beginning and ending at a port stop.

In order to support the cruise leg or 'commonality of collection platform' relationship, the library scheme provides a ship table data structure similar to the bounds table described earlier. Besides maintaining a pointer to the first position of each cruise, the ship table contains information about the cruise, including the unique cruise name, start and end dates, chief scientist's name, number of points collected, project name and funding agency. This information is used to satisfy ad hoc requests by researchers and management. Figure 6 is a diagram of the ship table.

A straightforward hashing algorithm (Knuth, 1973; Serverance and Duhne, 1976) is used to store and retrieve information from the ship table. This provides direct access to the table based on cruise identifier and is independent of the particular approach taken to identify the collection platform. As long as the table remains less than eighty percent full, quick access of this information is almost guaranteed.

Figure 6

Ship Table

Record 1	First pointer to alphabetical linked list, modulo value used in hashing address computation, next free record in second part of table.
Record 2 Record # 'Modulo'	These records are accessed directly using the cruise leg identifier in a hashing algorithm. Each record contains the identifier, next pointer in the alphabetical linked list, next pointer for cruise leg details and security code.
	Each cruise leg which is identified in the first part of the table maintains its own separate linked list of records further describing itself. This second part of the table stores the following information about each cruise leg: port stops, chief scientist's name, project name, start and end dates, comments, contributor, the number of data points for each data type stored for the cruise and pointer into the primary data file.

I considered other approaches in implementing the geographical proximity and cruise leg organization relationships. For example, the information contained in the bounds table and the ship table could have been placed together in one data structure. This method would save disk storage and simplify data retrieval for certain types of user queries. However, an important disadvantage is that there is a loss of simplicity. Adding new relationships would overly complicate this combined data structure.

Dictionary Structure

The dictionary describes the attributes of each data element stored in the library. Using a dictionary allows for easier growth when new data attributes are added, as well as fostering data integrity. This latter aspect is accomplished by including valid range values in the dictionary and requiring each data attribute to be in this range before it is stored. The dictionary also provides the mechanism to reduce data storage. It contains the information necessary to decode stored, compressed data. A hash addressing technique provides near $O(1)$ access to entries in the table.[1] Collisions are handled using the primary area overflow technique (Knuth, 1973).

---

[1]The notation $O(f(n))$ is read "of order f of n". The meaning here is that data access depends on a function of n, the number of records in the data structure. With $f(n)=1$, a constant, data access is independent of the number of records in the data structure.

When an item description of a new data parameter is added to the dictionary, it is assigned a unique prime number. The number is then used to help determine what data attributes are available at each geographical location without having to access these data individually. This is accomplished by maintaining a product of prime numbers at each geographical location. A simple division and check for remainder determines the existence of data at a location. For example, assume that the 'data available' product for the location under consideration has the value 782 (the product of the prime numbers 2, 17 and 23). A user wants to know if 'free-air anomaly' and 'geomagnetic anomaly' values exist at this location. The dictionary may have assigned the prime numbers 23 to 'free-air anomaly' and 19 to 'geomagnetic anomaly'. The retrieval software now performs two 'division and check for remainder' operations. Since 782 divided by 23 yields 34 and no remainder then a 'free-air anomaly' value has been stored and is available. However, 782 divided by 19 yields 41 with a non-zero remainder of 3. This means that a 'geomagnetic anomaly' value has not been stored and is unavailable for retrieval. This procedure eliminates the need for the system to actually access secondary storage to check if the requested data exists.

Data Storage Scheme

I have classified information to be stored in the library into three categories: primary data, secondary data and pointers. In this section I describe how these data are organized to provide an efficient, cost effective and flexible means to store location-dependent data.

The primary data include date, time, latitude and longitude information. These data are organized into direct access files according to the particular one degree geographical square in which the data reside. Primary data are stored in primary data files. I selected this approach as it provides:

1. growth potential for other than geophysical data,
2. the ability to reduce the on-line size of the library while affecting a minimum number of users,
3. economic data storage, and
4. simplicity.

One implementation of this scheme would require as many as 64,800 files if data existed in each one degree area of the world. Even taking into account my relatively sparse data set, this would be too many files for the operating system. However, in the implementation selected, only four files are used to cover the world, one file for each geographical quadrant of the earth. I define the four geographical quadrants as the northeast, northwest, southeast and southwest portions of the earth's surface. With this definition, Canada is located in the northwest quadrant and Australia is located in the southeast quadrant. Within each quadrant file, data can still be accessed by degree because of the bounds table structure described earlier. This scheme retains the desired retrieval capabilities without undue file manipulation.

Other divisions of the data can be implemented, if the algorithm which converts a position into a file name is changed. I retain this flexibility by using a direct encoding of the latitude and longitude values to represent a file name. A side benefit of using an encoding scheme rather than store the file names explicitly is that less storage space is used.

All information about a geographical location not considered as primary data is called secondary data and is stored in a separate area, the secondary data files. Among the many advantages to this approach are the following:

1. Adding new data attributes does not effect the organization of the primary data.
2. Secondary data can be removed entirely from on-line disk storage without affecting queries based on position or time.

3. Selected portions of the secondary data can migrate on and off the disk as the needs for these data change.

4. The data storage cell size need not be the same in the primary and secondary files, making better use of disk space.

Although the grouping of data within a storage cell (consecutive words of disk storage) in the secondary data files can be arbitrary, data which are known to be retrieved together are placed in the same storage cell to improve retrieval times.

The third category of information is pointer data. Pointers are used to link related information, using a linked list structure. Pointer fields in the primary data files exist for each relationship in which position records may participate. Currently, these relationships are the geographical proximity relation and the cruise leg organization of the data. A linked list structure is also used in the secondary data files to connect data attributes acquired at the same location and time.

I chose this approach to link information because:

1. new data attributes can be easily added;
2. newly identified relationships can be implemented easily by adding an additional pointer between common entities;
3. data can be easily deleted from a linked list; and
4. lists can be reordered to improve retrieval.

Functional Components

This section describes major functional components of the library system: query commands; use statistics, interface to existing Mercator chart making and analysis routines, and data security and integrity.

The query commands used to access the library take the following form

```
            !--      --!       !--!--     --! --!
command verb!/option   !  ...  !  !object   !,  !
            !__      __!       !__!__     __! __!
```

where 'command verb' is an acceptable command action verb such as DISPLAY, DEFINE, HELP, LIST, SHOW, and SET; 'option' is zero, one, or more verb modifiers separated by a slash; and object is zero, one, or more parameters separated by a comma. Appendix 4 is a complete user's manual and describes these commands in detail.

Figure 7 shows a sample query session. Once the user is logged onto the computer, the query program is initiated with a RUN DBAO:[GNG]DBQUERY command. The program uses two consecutive "greater than" symbols as a prompt to the user. To improve readability, all input entered by the user is underlined here.

The first command shown is a LIST/FULL command. This command causes the system to retrieve all the information it has about the specified cruise leg. Note that a summary of the data available for the cruise is included. This information is readily available from the ship table and involves a simple direct lookup.

The second command shown is the HELP command, used here to provide instructions as to use of the DEFINE command. The HELP facility provides a user with basic information about how to use the retrieval program. The third command shown is the DEFINE command, used here to obtain the definition and certain attributes of the dictionary term 'corrected depth'. Note that this same dictionary structure is used by the data insertion software to foster data integrity by verifying the type and magnitude of values being inserted. The DEFINE command can be used with the ALL option to obtain an alphabetical list of all valid dictionary terms.

The fourth command shown is the SET command, which restricts the user's area of interest to the geographical area bounded by 21 degrees south by 25 degrees south and 14 degrees east by 20 degrees east. The system saves this user-supplied information internally for use during retrieval. The fifth command, DISPLAY, results in a summary of the data parameters available within these bounds. To satify this request, the system searches each bounds table affected by the user-selected geographical bounds and computes a sum of all data parameters available. Finally, the user ends the query session by issuing the END command. The system responds with a message and a summary of the computer resources used during the session. Additional examples of retrieval commands are included in the user's manual, Appendix IV.

Figure 7

Sample Query Session

```
$ RUN DBAO:[GNG]DBQUERY

          DBQUERY - VERSION 1.00 - 80OCT29 08:12:21

     HI! With this program you can request information
     from the dictionary and library files.  If you need
     assistance with any of the commands just type in HELP
     for more information.

>> LIST/FULL ATLANTIS II 67 5

*OTPTSTBLF message, full contents of ship table entry follows:

Cruise leg id: ATLANTIS II 67 5
   Start date: 1972/ 4/12    0.00  End date: 1972/ 5/ 6  0.00
         Date added: 1980/ 5/13 1617.54
   Flags:      0  Security code: 0  Reference #:        151
         Reserved:     0
     Project name:  IDOE
     Chief scientist:  MILLIMAN
     Port stops:  WALVIS BAY-WALVIS BAY
     Contributor:  Woods Hole Oceanographic Institution
     Comments:  SE ATLANTIC OCEAN

      Parameter name                    Number of Data Points

 CORRECTED DEPTH                                1310
 POSITION(S)                                    1310

 >>HELP DEFINE

    DEFINE Command:

    This command allows you to retrieve dictionary term
    definitions and other dictionary information from the
    dictionary.  The form of the command is

              DEFINE/OPTION  [TERM1],[TERM2],...

    where [TERM1], [TERM2], etc. are terms that you wish to
    have defined.  If the terms are not in the dictionary a
    message will indicate that.
```

Figure 7 (continued)

```
    OPTIONS:

                /ALL      generates an alphabetical list of all
                          dictionary terms.  You may restrict
                          this list by specifying the starting
                          and ending terms.
                /PARTIAL  Display definition plus some more
                          information
                /WORD     Display only the definition (default)

>>DEFINE/PARTIAL CORRECTED DEPTH

CORRECTED DEPTH:
    Water depth (   0) corrected for transducer depth, sound
    velocity, and tides

    ID #:      19    Data file Type:  secondary
    Starting byte position in group # 2 is 9 (Bit Position:  0)
    Original variable type:  real
    Variable stored as type:  integer*4

>>SET/BOUNDS -21,-25,14,20

>>DISPLAY

*PRTDATSUM message, summary of          -21
  data available in the area          --------
  indicated to the right.             !      !
                                   14 !      !  20
                                      !      !
                                      --------
                                        -25

      Parameter name                  Number of data points

 POSITION(S)                                    44
 CORRECTED DEPTH                                44
 GEOMAGNETIC ANOMALY                             0
 FREE AIR ANOMALY                                0

>>END

GOOD BYE FOR NOW.

TIMES IN SECONDS       PAGE      DIRECT     BUFFERED
    CPU  ELAPSED      FAULTS      I/O         I/O
   2.63   51.28        504        151         118
```

A menu approach for the user interface was also considered but not implemented. A high-speed cathode ray tube type terminal is essential for this approach and I anticipated that many users would be annoyed when they had to use slower speed terminals. On-line documentation is provided through the use of the HELP command.

The system has a record-keeping facility to keep use statistics. This facility provides a mechanism for documenting the library activity and is useful for the following reasons:

1. Unused, secondary information can be removed from on-line storage to reduce storage costs.
2. Charges based on actual use can be computed.
3. Library growth and data retrieval patterns can be studied.
4. The need for data reorganization can be determined.

The interface to data analysis software, especially the chart plotting programs, will be implemented using a Fortran routine which allows another program to retrieve data directly from the library based on geographical boundaries and cruise. The routine passes back to the calling program one primary record at a time and zero or more secondary data attributes depending on what was requested.

The more common retrieval requests, such as requests for navigation or data charts, will be part of the query software. Users can specify the chart bounds, scale and parameters needed by the plotting routines. These parameters are then reformatted and sent to an existing plotting routine.

It is also possible to produce copies of data from the library in the exchange format (NOAA, 1977) on tape. This option allows for simple data exchanges among other research and government organizations and also allows users to have their own copies of the data.

Data integrity is maintained by the use of parameters stored in the dictionary to verify all data before they are stored in the library. Recovery from software or system crashes is accomplished by rolling back the library to the state it was in before the event. This is accomplished by maintaining a copy of all data base files on magnetic tapes. Data security is provided by an authorization file which maintains information about valid users. Data insertion, modification, or deletion can only be performed by authorized individuals.

Chapter V

Conclusion

I have designed and implemented a storage and retrieval scheme which provides cost effective and easy access to location-dependent, 'geophysical' data. The scheme is operational on a Digital Equipment Corporation VAX-11/780 computer. Information about data, such as port stops, project name and funding agency, as well as data values, are available, on-line, to a time sharing user validated to use the system. Appendix IV is a user's manual which describes how to use this system.

In order to minimize the software development time and effort, I took advantage of the capabilities of the operating system and existing software whenever possible. For example, standard system supplied utilities are used to back up the library on magnetic tape and existing graphics routines were used. However, even with these shortcuts, the software system takes more than 14,000 lines of code. Appendix V contains a list of system routines by category and some software statistics.

These computer routines do not quite comprise a true data base management system since they do not provide users with different logical views of the data. However, other features of a DBMS are present: multi-user access to the data is provided; data requests are satisfied with reasonable speed; storing and retrieving costs are minimized; and, data are stored accurately and consistently.

This new data storage and retrieval system can be considered a success only if it is used, and if it provides the features needed by at least a majority of the users. Demonstrations of the system have been well received. The full power of the system will not be realized until a significant portion of existing data are added. I anticipate that this task will be complete within a few months.

Implementation of the system has followed the original design plans. Certain detail features, missed during the early planning but evident during implementation and testing, were added. For example, the dictionary needed to store the original format of a data parameter, as well as the stored format, so that values can be converted back from their compressed form into their original form. Also, I changed the bounds table by including summary data statistics. This feature was not included in the early design, but is needed to facilitate user's requests about the existence of data values.

Finally, implementing the data insertion routine was more difficult than expected. Manipulating the data and bounds tables as the data enters and leaves degree squares was quite complex.[1] However, even with the extensive data processing, the central processing time needed to store data in the new scheme compares quite favorably to the time to store data in the older, sequentially accessed scheme. The improved accessibility to information about the data and cruise legs, as well as the improved accessibility to the data values, has been achieved.

---

1 Part of this difficulty was due to the decision to add to the primary and secondary data files sequentially, rather than randomly. This design feature minimizes the use of disk space and improves processing speed but is at the expense of programming complexity.

Appendix I

Data Attributes

ALTITUDE MAGNETIC SENSOR

Position of the primary magnetic sensor below (-) or above (+) sea level, in meters. Synonym: DEPTH MAGNETIC SENSOR

BATHYMETRY CORRECTION CODE

Details the procedure used for determining the sound velocity correction used to correct the water depth measurement.

BATHYMETRY QUALITY CODE

Specifies the quality of the recorded depth measurements.

BATHYMETRY TYPE CODE

Specifies how the depth value was derived (e.g. observed, interpolated, etc.).

BOUGUER ANOMALY

A specific interpretation of the gravity free-air anomaly taking the water depth into account, in milligals.

CORRECTED DEPTH

Water depth corrected for transducer depth, sound velocity, tides, etc., in meters.

CRUISE

Identifies a particular voyage of a ship in which the ship made one or more port stops. Synonym: cruise number.

CRUISE IDENTIFIER

Identifies a particular cruise leg or legs of a voyage. This is usually unique within a particular data gathering agency. Synonym: a concatenation of SHIP, CRUISE and LEG.

CUMULATIVE DISTANCE

The distance traveled by the ship since the start of the cruise leg (usually with 0 at the departing port) until the present time, in kilometers.

CURRENT HEADING

The direction or orientation of the water current, in degrees. This value is derivable from the CURRENT SPEED NORTH and CURRENT SPEED EAST.

CURRENT SPEED EAST

The component of the current velocity in the east (positive) or west (negative) direction, in meters per second.

CURRENT SPEED NORTH

The component of the current velocity in the north (positive) or south (negative) direction, in meters per second.

CURRENT VELOCITY

Vector specifies the speed and direction of the current through which the ship is traveling. This value is usually given as two numbers, the north and east components or as the *current speed and current heading*.

DATA RECORD TYPE

Under existing data storage schemes this parameter identifies the type of record in order to distinguish among header and data record types. In a new scheme it can have a similar meaning and takes on a more central and important role during data retrieval.

DAY

Specifies the day of the month. When combined with the month, a form of Julian day is formed which can be combined with the hour. See HOUR DAY.

DEPTH MAGNETIC SENSOR

See ALTITUDE MAGNETIC SENSOR

DIURNAL CORRECTION

A correction applied to the measured magnetic field in order to correct for daily variation in the regional field.

EOTVOS CORRECTION

A correction applied to the measured total gravity field related to the ship's speed and heading, in milligals.

FREE-AIR ANOMALY

The observed gravity minus the theoretical gravity value, in milligals.

GEOMAGNETIC ANOMALY

The total geomagnetic field minus the theoretical value for the total field. The diurnal correction has been applied if available, in gammas. Synonyms: magnetic anomaly, magnetic residual field, total geomagnetic residual field.

GEOMAGNETIC QUALITY CODE

Specifies the quality of the recorded geomagnetic measurements.

GRAVITY QUALITY CODE

Specifies the quality of the recorded gravity measurement.

HEIGHT

The height above sea level (plus) or below sea level (negative) of the acquisition platform. For ship acquisition this is usually taken to be 0, in kilometers.

HOUR DAY

An encoding of the hour within the day and the day of the year, with January 1st being day 1, defined as HOUR * 1000 + day of year.

LEG

Identifies the particular port-to-port segment of a cruise. Synonym: leg number.

LATITUDE

The ship's position in the north (positive) - south (negative) direction on the earth's surface, in degrees.

LOCATION

The position of the data collection platform, usually defined in terms of latitude, longitude and height. Synonym: position.

LONGITUDE

The ship's position in the east (positive) - west (negative) direction on the earth's surface, in degrees.

MAGNETIC FIELD SENSOR

Specifies which recording sensor was used as the primary recording instrument.

MINUTE

Specifies the minute and fraction of the hour, in minutes.

MONTH

The month during which a data point was recorded. This value is encoded along with the day of the month, into a Julian day. Julian day is further combined with the hour of the day. See HOUR DAY.

NAVIGATION QUALITY CODE

Specifies the quality of the recorded navigational measurements.

NAVIGATION TYPE CODE

Indicates how the navigational information was obtained.

OBSERVED GRAVITY

Measured total gravity measurement, corrected for Eotvos, drift and tares in milligals.

PROTECTION CODE

Specifies the access privilege required by the user in order to retrieve data.

SEISMIC SHOT POINT

Identifies the shot points for single and multichannel seismic data so that this data can be analyzed with other underway measurements.

SHIP

Identifies the platform from which data were collected. Synonyms: ship name, cruise ID, platform. Note: Some institutions concatenate ship, cruise, and leg into a single unique identifier and refer to this entity as the cruise ID.

SHIP HEADING

The instantaneous orientation of the ship, in degrees. See SHIP VELOCITY.

SHIP VELOCITY

The instantaneous ship's speed north (+)/south (-), and speed east (+)/west (-) from the current ship's position to the next measured instant of time, in meters per second. See SPEED NORTH and SPEED EAST.

SPEED EAST

The instantaneous ship's speed in the east (plus) and west (negative) direction, a component of the ship velocity, in meters per second.

SPEED NORTH

The instantaneous ship's speed in the north (plus) and south (negative) direction, a component of the ship velocity, in meters per second.

TIME

The time of the data measurement(s), in GMT. See HOUR DAY and MINUTE.

TIME ZONE

Usually the time of a data measurement is recorded in Greenwich Mean Time (GMT) even though the ship operates in local time. Time zone is zero if the recorded time is GMT, else non-zero.

TOTAL MAGNETIC FIELD-1

The value of the total geomagnetic field measured by the primary sensor, in gammas (or nanotesla). Synonyms: total mag field, total mag field 1, geomagnetic field, total geomagnetic field.

UNCORRECTED DEPTH

The water depth, historically recorded in fathoms with an assumed sound speed of 800 fathoms per second (round trip travel time), in seconds assuming sound speed of 800 fathoms per second.

YEAR

The year that a measurement was collected.

Appendix II

General Design Specifications

This section is a copy of the original system design specifications. However, not all features described here are currently implemented. In particular, the following features are now available: inserting new data in the library, data summaries by area, and data integrity features including a record of transactions and verification of data parameters before they are stored. Features still under development include data value retrieval by cruise leg and geographical area and the interface to existing charting and profiling graphics routines.

I. Loading the library.

A. Data insertions: underway geophysical data including navigation, bathymetry, geomagnetics and gravity, are supplied to the library in the required merged or merged-merged format.

B. Data deletion: data in the library which is in error can be deleted one 'cruise leg' at a time.

C. Data update: a 'cruise leg' whose navigational information must be changed is first deleted from storage. The corrected data are then added as a standard insert operation. If individual measurements within a 'cruise leg' are in error, and do not involve positional elements, then these measurements can be individually replaced.

II. Retrieval.

A. Data retrieval (to hard-copy plots, listings or magnetic tape files) is accomplished by specifying:

1. Specific cruises.
2. Specific geographical area.
3. Date/time pairs.
4. Any combination of (1) through (3).

B. The system supports interrogation of the library in order to provide answers to the following forms of queries:

1. How much data exists in the North Atlantic Ocean?
2. How much data exists in the geographic area bounded by 10 degrees north, 5 degrees south, 35 degrees west and 5 degrees east?
3. Does data exist for ATLANTIS II cruise 17 leg 6?

4. Display on my CRT screen the ship's tracks located in the following area: 25N, 20N, 30E, and 5E, collected since 1975.
5. Contour the 'bathymetry' data located in the 2 degree square surrounding the point 30 degrees south, 5 degrees east; and display the contour plot on my CRT screen.

C. The system supports the following forms of graphical display options:

1. Annotated charts displaying cruise tracks and data values along the ship's track.
2. Profiles of selected cruise leg data versus time or cumulative distance.
3. Profiles of selected cruise leg data, plotted at right angles to the cruise track.
4. Contour plots of any data parameter.

III. Data integrity.

A. A chronological record is kept of all transactions with the library involving insertion, deletion or replacement, including the number and kinds of records involved.

B. The ability to recover from catastrophic failures such as power loss during the insertion/deletion/replacement operation is provided.

IV. Operation, maintenance and cost considerations.

A. Information is maintained by the system so that each user of the library can be charged according to the use of the system.

B. An activity summary is kept of the retrieval use of the system, including the number of users, the types of requests, the number of records retrieved, and the computer resources used, such as elapsed and cpu time.

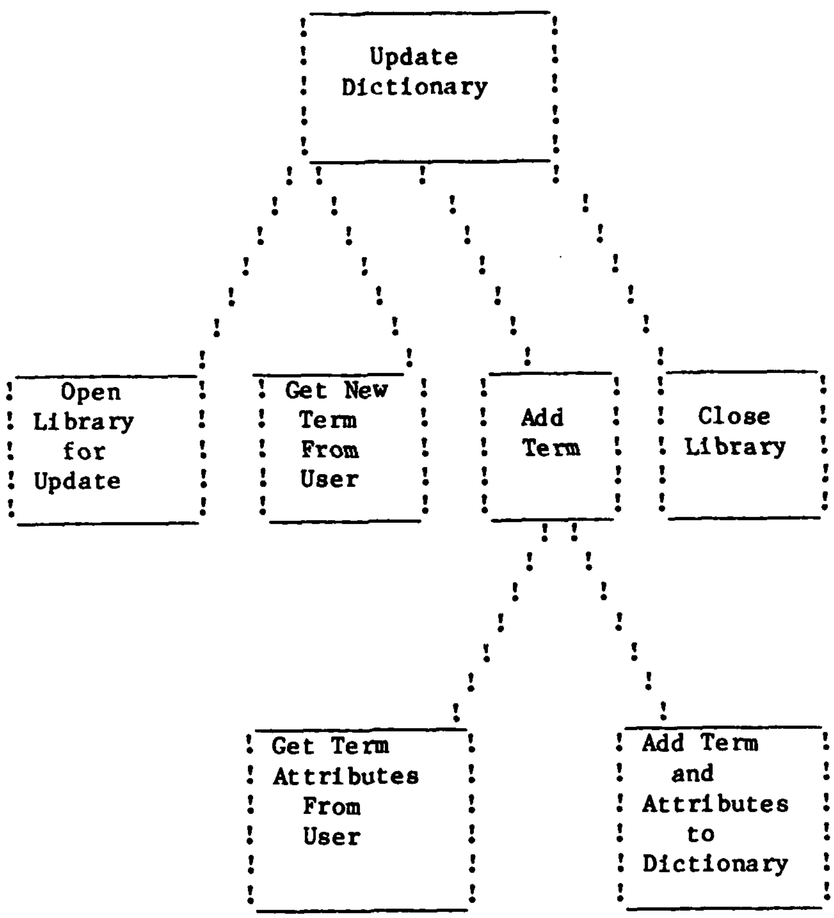
C. In order to support full on-line access to the entire library a significant portion of a dedicated disk pack and drive is required.

Appendix III

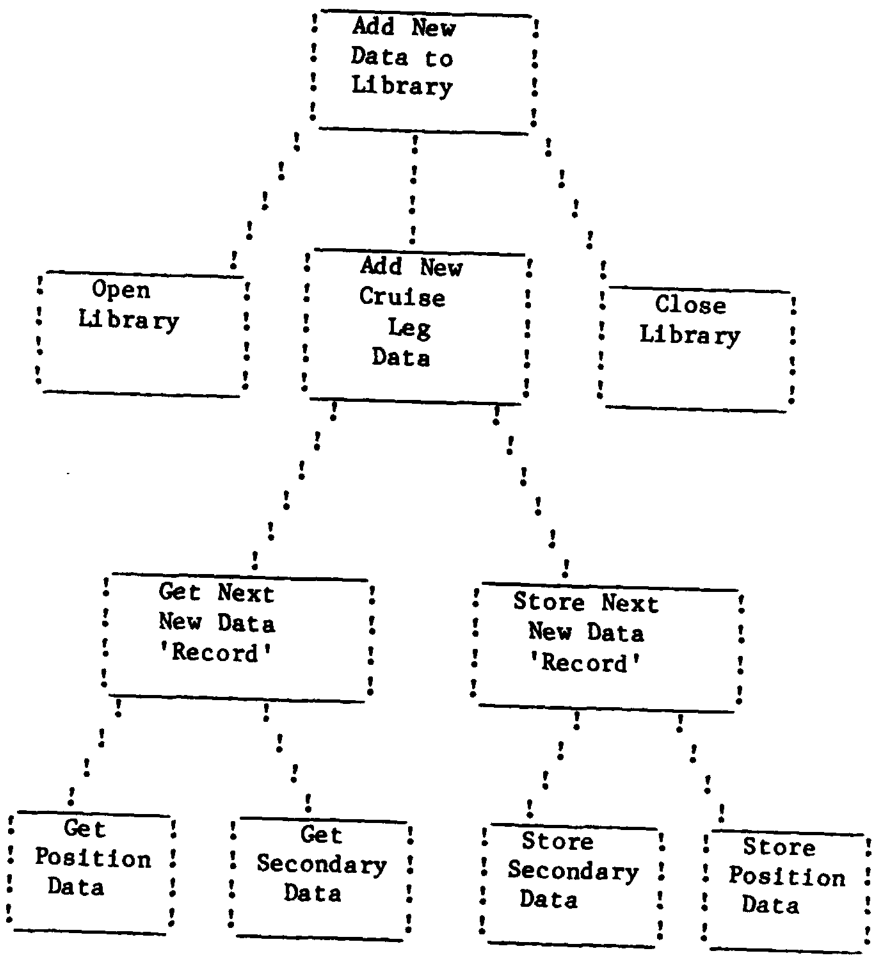
Program Structure Diagrams

This section displays tree structured program organization diagrams for three important functions in the library system: dictionary update, data insertion and user query.

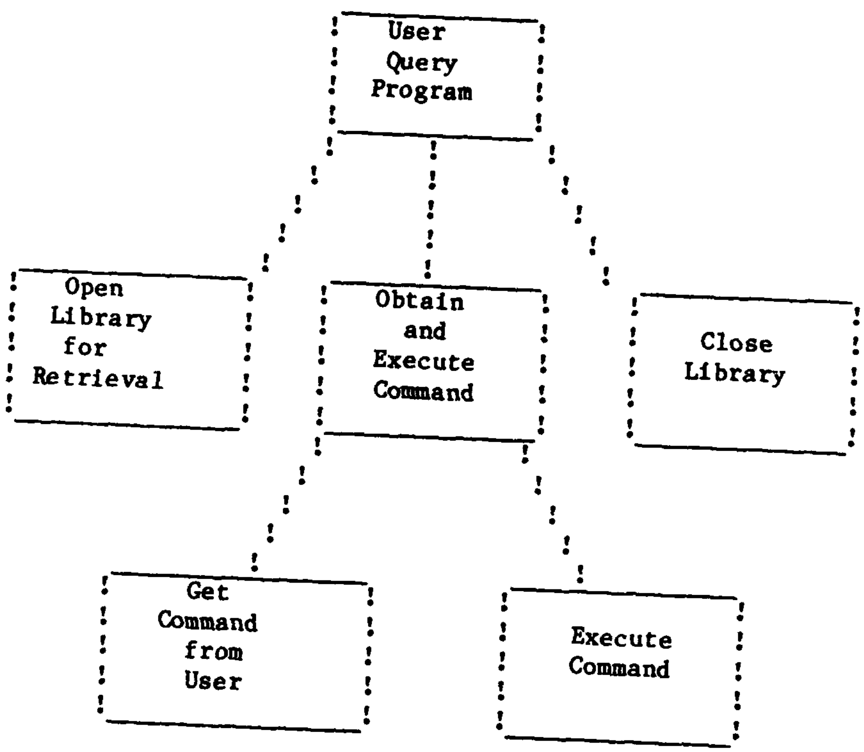
Dictionary Update



Data Insertion



User Query


User Query Program
Open Library for Retrieval
Obtain and Execute Command
Close Library
Get Command from User
Execute Command

Appendix IV

User's Manual

This appendix is the user's manual for DBQUERY, the data retrieval program for position-dependent data. The manual describes how a user at a time-sharing terminal connected to Woods Hole Oceanographic Institution's VAX-11/780 computer can access information about 'geophysical' data stored by the computer.

I. Capabilities

The program used to access information in the library for location-dependent data is called DBQUERY. This program allows the user to obtain information about 'cruise legs', retrieve data from the library, generate plots, and retrieve information about the parameters stored in the library.

II. Logging On

The first step that a library user must take is to connect the time-sharing terminal to the Woods Hole Oceanographic Institution's VAX-11 computer. If the user is outside of Woods Hole, then dial (617) 540-6000 and ask the operator for one of the following VAX-11 computer extensions depending on the transmission speed you require:

Extension Number	Baud Rate of User's Terminal
6600	300 band
6500	1200 band

If you have access to a W.H.O.I. "blue phone" you can dial one of these extensions yourself. (Some terminals at W.H.O.I. are permanently connected to the VAX and do not require a phone connection). Place your phone in the modem cradle when you hear the high pitched "carrier" tone.

At this point, touch the "return" key at your terminal. The computer will respond with

Username:

Enter your VAX user name followed by a return. If you do not have a user name then you must call the computer center and request one; after a valid user name has been entered the computer will print

Password:

Enter your password followed by a return. If you have entered these items correctly, the system will respond with

Welcome to VAX/VMS Version 1.60 (date)

III. Program DBQUERY Initiation

Once you have successfully logged onto the VAX and receive the dollar sign ($) prompt character you can begin using the query program by typing the following command:

RUN DBAO:[GNG]DBQUERY

Program DBQUERY will begin and print the following

DBQUERY - Version 1.00 - (today's date)

Hi! With this program you can request information from the dictionary and library files. If you need assistance with any of the commands, just type HELP for more information.

The user prompt for program DBQUERY is " ". That is, when the query program is ready to receive your next command it will display these characters. The following section describes the commands you can use. However, if you have not been authorized to use the library system you will get the following message:

*DBQUERY message, could not open the library for retrieval. Sorry!

If you get this message you cannot use the library. Contact Woods Hole's Digital Data Library staff at 548-1400 Ext. 2581 for authorization or if you have any other questions.

IV. User Commands

There are thirteen basic commands that you can issue to program DBQUERY. These are

COPY	HELP
DEFINE	LIST
DISPLAY	PLOT
DONE	SET
DRAW	SHOW
END	STOP

Commands take the following general form

```
            !--      --!   !--!--    --!---!
command verb!/option   !...!  !object  !,...!
            !__      __!   !__!__    __!___!
```

where 'command verb' is any one of the above commands; '/option' is zero, one or more command verb options separated by a slash; and 'object' is zero, one or more parameters used by the command, separated by a comma. Command verbs and options can be abbreviated to as few as two letters (e.g. CO for copy, HE for help and ST for stop).

A description of each of the basic comannds follows. They are listed in alphabetical order.

@ command:

This command allows you to execute one or more library commands that have been stored in a disk file. The file must already exist. At the present time all commands will be honored except another @ command.

The form of the command is

@ filespec

where 'filespec' is the name of a VAX/VMS file created either with the editor or other compatible utilities.

OPTIONS:

/ECHO print the commands read from filespec on the user's terminal

COPY command:

This command allows you to copy selected portions of the library onto a disk file. The specific data that you wish to copy are selected using the SET command.

The form of the command is

COPY

The default name of the file created is made up of the first 8 nonblank characters of your logon name with an extent of .DAT. Also, the SET command allows you to change the default output file name.

DEFINE command:

This command allows you to retrieve dictionary term definitions and other information from the dictionary. The form of the command is

DEFINE/option [term1],[term2],...

where [term1], [term2], etc. are terms that you wish to have defined. If the terms are not in the dictionary a message will indicate that.

OPTIONS:

/ALL	generates an alphabetical list of all dictionary terms
/PARTIAL	display definition plus additional information about where and how the parameter is stored in the library
/WORD	display only the definition (default)

Example

    DEFINE/PARTIAL FREE AIR ANOMALY

FREE AIR ANOMALY:
    observed gravity, corrected for Eotvos and drift, minus the
    theoretical gravity value

    ID #:   29   Data file type: secondary
    Starting byte position in group #   2 is   15 (Bit position: 0)
    Original variable type: real
    Variable stored as type: integer* 2

DISPLAY command:

This command prints a summary of the data which satisfy the current retrieval request as defined by the SET command. The form of the command is

    DISPLAY/option

Options:

/CRUISE_LEGS   lists the cruise legs with a summary of the number of measurements satisfying your retrieval specifications.

/DATA_SUMMARY lists the total number of measurements satifying your bounds and data type retrieval specifications. (Default)

Example

DISPLAY

*PRTDATSUM message, summary of data available in the area indicated to the right.

```
        -21
     --------
     !      !
  14 !      !  20
     !      !
     --------
        -25
```

Parameter name	Number of data points
POSITION(S)	44
CORRECTED DEPTH	44
GEOMAGNETIC ANOMALY	0
FREE AIR ANOMALY	0

DONE command:

This command stops the DBQUERY program and returns you to the VAX/VMS monitor.

DRAW command:

This command allows you to generate charts and profiles of data at your CRT screen, in real time. However, at this time the option has not been implemented.

END command:

This command stops the DBQUERY program and returns you to the VAX/VMS monitor.

Example

```
     END

GOOD BYE FOR NOW.

TIMES IN SECONDS     PAGE     DIRECT    BUFFERED
    CPU  ELAPSED    FAULTS     I/O        I/O
   1.74   162.17       159      143          76
```

HELP command:

The following library query commands are recognized:

```
     COPY  DEFINE  DISPLAY  DONE  DRAW  END  LIST  PLOT  SET

     SHOW  STOP  @
```

For more information about these commands enter

```
     HELP [command]
```

where [command] is any one of the above commands. Unique abbreviations for the commands and options, down to two letters, are allowed. For example, to obtain more information on the DISPLAY command any one of the following lines are allowed:

HELP DISPLAY

HE DISP

LIST command:

This command allows you to retrieve information about one or more cruises from the library ship table. The form of the command is

LIST/option cruise_leg_1,cruise_leg_2,...

where cruise_leg_1, cruise_leg_2 etc. are the names of the cruise legs.

OPTIONS:

/ALL produces an alphabetical list of all the 'cruise legs' available in the library

/BRIEF	prints some information about the 'cruise legs' listed in the command, such as the start and end dates, project name, port stops, chief scientist and contributor. (Default)
/FULL	prints all the information available about the 'cruise legs' listed in the command.
/MGD77	prints the MGD77 header records of the 'cruise legs' listed in the command

Example

```
LIST/FULL ATLANTIS II

*OTPTSTBLF message, full contents of ship table entry follows:

Cruise leg id: ATLANTIS II 67 5
   Start date: 1972/ 4/12    0.00  End date: 1972/ 5/ 6    0.00
              Date added: 1980/ 5/13 1617.54
   Flags:   0  Security code: 0  Reference #:   151  Reserved:   0
     Project name:  IDOE
     Chief scientist:  MILLIMAN
     Port stops:  WALVIS BAY-WALVIS BAY
     Contributor:  Woods Hole Oceanographic Institution
     Comments:  SE ATLANTIC OCEAN

      Parameter name                   Number of Data Points

 CORRECTED DEPTH                                1310
 POSITION(S)                                    1310
```

PLOT command:

This command allows you to generate hard copy plots of charts, profiles and contour maps. Currently, however, this option is not implemented.

SET command:

This command allows you to restrict or further define your data retrieval request. The form of the command is

SET/option parameter_1,parameter_2,...

Each option calls for a different number of parameters. Consult the options list below to determine the form and number of parameters to include.

Example

SET/BOUNDS -21,-25,14,20

Options:

/BOUNDS restricts the retrieval boundaries to the specified top, bottom, left and right sides of a geographical rectangle. The default is the whole world. For example, to retrieve data which lie only within the area bounded by 10 degrees north, 30 degrees south, 14 degrees west and 5 degrees east enter

```
SET/BOUNDS 10,-30,-14,5
```

Note that the sign convention is south and west negative. (Default is the entire world.)

/COPY_OUTPUT change the output file or device name for the COPY command to the specified file name. The default file name is made up of the first 8 non-blank letters of the user's name with the extent of .DAT. For example, to change the file name to MYDATA.DAT enter

```
SET/COPY_OUTPUT MYDATA.DAT
```

To return to the default output file name enter

SET/COPY_OUTPUT

/CRUISE_LEG restricts the retrieval to the cruise legs specified in the rest of the command. The default is ALL of the cruise legs. For example, to restrict retrieval to data from CHAIN 115 leg 3 and ATLANTIS II 67 leg 2 enter

SET/CRUISE_LEG CHAIN 115 3,ATLANTIS II 67 2

/DATA_TYPE restricts retrieval to the data types specified. For example

SET/DATA_TYPE BOUGUER ANOMALY

will result in retrieving the Bouguer anomaly data parameter. The default data types are corrected depth, geomagnetic anomaly and free air anomaly. To return to the default data types enter

SET

To retrieve all data, enter

SET/DATA_TYPE ALL

SHOW command:

This command allows you to display the current values for retrieval options selected either by default or explicitly through the use of the SET command. The form of the command is

SHOW/option

Option

/BOUNDS	describes the geographical retrieval boundary. (Default)
/COPY_OUTPUT	gives the name of the disk file or device which will receive the output from a COPY command
/CRUISE_LEG	lists the 'cruise legs' from which retrieval is restricted
/DATA_TYPE	lists the data types that will be retrieved

STOP command:

This command stops the DBQUERY program and returns you to the VAX/VMS monitor.

Appendix V

Computer Routine Descriptions

This appendix provides a summary listing by category of the computer routines which I designed and wrote to implement the data library. There currently are a total of 314 routines which make up the software for the data library system. With a total of 14232 lines of code, this yields an average of about 45 lines per routine. Sixty-six percent of the lines are executable code while the rest are comment (documentation) lines.

Main Programs

DBINSERT -- insert new data for an entire cruise leg

DBQUERY -- library query and retrieval program

DBVALID -- validate a new user for the library system

INITDBLD -- initialize the data dictionary, history file, validation file and ship table

SHPTBLINS -- add new cruise leg information to the ship table

UPDBDIC -- update the library dictionary with a new data attribute

Input/Output routines

CLSDTLAPN	OPNRWPRI	RDSTBL5
CLSPRIAPN	OPNRWSTBL	RDSTBL7
DECLOSE	OPNWVLDE	RDSTBLCTS
DPOPNEXEC	OTPTDBMIS	RDSTBLDT
DBOPNRTV	OTPTSTBLF	RDVLDF
DBOPNUPD	PRTCNTS	READDICR
GETDDLNAV	PRTSTBL	READFULLT
GETNEWNAV	PRTTRMALL	READSEAG
GETNEWREL	PRTTRMDEF	READSTBLR
GETUSRCMD	PRTTRMPAR	READSTSO
INPTATBTS	PUTNEWDAT	RPBNDL
INPTINTP	PUTNEWDTL	RSTBLPAR
INPTSYSP	RDBND2	WRTDBHIS
INPTUSRP	RDBNDPAR	WRTDIC2
INSTODB	RDOIC2	WRTDIC23
OPNDBHIS	RDDIC23	WRTDIC3
OPNDTLAPN	RDOIC5	WRTDIC5
OPNPRIAPN	RDOICDEF	WRIDICr
OPNRDIC	RDICPARAM	WRTINTDT6
OPNRPRI	RDSTBL11	WRTFYKKT
OPNRSTBL	RDSTBL13	WRTSTBL11
OPNRWBND	RDSTBL19	WRTSTBL17
OPNRDDIC	RDSTBL2	WRTSTBL19
OPNRDDTL	RDSTBL3	WRTSTBL2

Input/Output routines (continued)

WRTSTBL3

WRTSTBL5

WRTSTBL7

WRTSTBLDT

WRTSTBLR

WRTSYSP

WSTBLPAR

WTBND2

WTBND3

WTBNDPAR

WTDICDEF

WTDICUNTS

WTNEWDTL

WTNEWNAV

WTSDTLR2

WTSPRIR1

WTSPRIR2

WTSPRIR3

WTVLDF

Specific functions

ADDCNTBND	DBQRYSET	EXPSET
ADDCNTPRI	DQBRYSHOW	EXPSHOW
ADDCRUISE	DEFBNDFIL	EXPSTOP
ADDINTOIC	DEFDBBND	FLODTOPOS
ADDNEWUSR	DEFBDDIC	FNDOICTRM
ADDSTBL	DEFDBDTL	FMDSTBLID
ADDTODIC	DEFDBHELP	GETDICOEF
ADDTOSTBL	DEFDBHIS	GETINTPAR
ALPHAPOS	DEFDBPRI	GETNEWBIT
ALPHASTBL	DEFDBVAL	GETNEWCHR
CHKUSREXE	DEFDTLFIL	GETNEWCPX
CHKUSRUPI	DEFNULVAL	GETNEWDBL
CODTOAFIL	DEFPRIFIL	GETNEWINT
COOTOPFIL	DEFSTBL	GETSTBLE
CPOSBNDE	DEFSYSP	GETVARID
DBACTION	DEFVARTYP	INITOBHIS
DBCMDOP	EXPATFL	INITOIC
DBMENUOPP	EXPDEFN	INIT STBL
DBQRYATEL	EXPDONE	INIT VLOF
DBQRYDEFN	EXPDRAW	MAKBNDFIL
DBQRYDRAW	EXPOSPY	MAKDTLFIL
DBQRYDSPY	EXPEND	MAKPRIFIL
DBQRYHELKP	EXPHFLKP	MVJINFSTG

Specific functions (continued)

DBQRYLIST

EXPLIST

POSTDAFIL

DBQRYPLOT

EXPPLOT

POSTOFCOD

POSTOFPFIL

SETQRYDFT

UPDDBDIC

UPDBNDR2

UPDBNDR3

UPDDBSET

UPDSTBL

UPDTPNTRS

UPTALPHA

UPTSTBLA

Utility routines

ALIMSSRNK	FNDSTRING	OBSGXPND
ALIMSXPND	GANOMSRNK	OPENERR
BLNKFILE	GANOMXPND	OPNSEQR
CDISTRSNK	GETCMPDAT	POSSANK
CDISTXPND	GETJINF	POSXPND
CHGNVMTIM	GETNXTPNO	PRSSWTCH
CLOSEFILE	GETUSRNAME	PRSVERB
CLOSERR	HASHTEXTADDRESS	READBTXT
COUNTERS	HSEARCH	READERR
DATESRNK	IDTOMULT	RECNOTFND
DATEXPND	INPTTXT	SRNKR4I2
DATBITSAM	LISTFILE	SRNKR4I4
DELAY	LREADERR	SUBQUAD
DEPTHSRNK	LSTNONBLK	TERMDESC
DEPTHXPND	LSTSTRING	TESTLL
DREADERR	LWRITERR	TMAGSRNK
DRNLCSRNK	MAKELL	TMAGXPND
DRNLCXPND	MANOMSRNK	TZONESANK
DWRITERR	MANOMXPND	TZONEXPND
EOFFOUND	MTNUMSRNK	UNXEDEOIF
ERRMESS	MTNUMXPND	UPDOATCNT
EOTVOSSRNK	NEXTPRIME	UPLSTDBR
EOTVOSXPND	NOROOM	VELSRNK

Utility routines (continued)

FNDDBREC

NXTBLK

VELXPND

FNDLSTREC

NXTNONBLD

WRITERR

FNDNXTFRE

OBSGSRNK

WRTDBTXT

XPNDI2I4

XPNDI4R4

XTRCTOBJ

ZCOORSRNK

ZCOORXPND

References

Alvarez, D.T. and Taylor M.L. 1974. Cartographic Data Base Hierarchy. vol. 1: Systems Analysis and Design (McLean: PRC Information Sci. Co.).

Amidon, E.L. and Akin, G.S. 1971. Algorithmic Selection of the Best Method for Compressing Map Data Strings. CACM. 14: 769-74.

Anderson, K. Eric. 1978. Spacial Analysis in a Data Base Environment in Harvard Papers on Geographic Information Systems, Edited by Geoffrey Dutton. Vol. 2. Boston: Harvard University.

Bender, Eric. 1979. A Quick Guide to Ocean Environmental Data Files. Sea Technology, 20 (2): 10-13.

Brooks, Frederick P. 1975. The Mythical Man Month. Reading: Addison-Wesley.

Chen, Tien Chi and Irving, T. Ho 1975. Storage-Efficient Representation of Decimal Data. CACM. 18: 49-52.

Chrisman, Nicholas 1977. Impact of Data Structure on Geographic Information Processing. Laboratory for Graphics and Spacial Analysis, Graduate School of Design, Harvard University, Internal Report no. 74-4.

Cook, Bruce G. 1978. The Structural and Algorithmic Basis of a Geographic Data Base, in Harvard Papers on Geographic Information Systems, Edited by Geoffrey Dutton, Vol. 4. Boston: Harvard University.

Cook, G.C. and Mayo, K.K. 1977. A Geographic Data Base Using FORDATA. Australian Computer Journal, 9: 145-149.

Dampney, C.N.G., Funkhouser, Don, and Alexander, Michael 1978. Report Structure of the SEG-Point Data Exchange and Field Formats (A Proposd SEG Digital Recording Standard) - Specification. Geophysics. 48: 216-227.

Date, C. J. 1977. An Introduction to Data Base Systems. Reading: Addison-Wesley.

AD-A116 767 WOODS HOLE OCEANOGRAPHIC INSTITUTION MA F/G 9/2
COMPUTER STORAGE AND RETRIEVAL OF POSITION - DEPENDENT DATA.(U)
JUN 82 R C GROMAN N00014-79-C-0071
UNCLASSIFIED WHOI-82-27 NL

2 OF 2
AD A
116 767

END
DATE
FILMED
6-82
DTIC

Davisson, Lee D., and Gray, Robert M., Eds. 1976. Data Compresssion. Benchmark Papers in Electrical Engineering and Computer Science, vol. 14, Dowden, Hutchinson and Ross, Inc.

Dishon, Yitzhak 1977. Data Compaction in Computer Systems. Computer Design Publishing Corp..

Ferguson, R.B., Maddox, J.H., and Wren, H.F. 1977. Data Management and Analysis Systems for Large-Scale Hydrogeochemical Reconnaissance. Computers and Geoscience. 3: 453-458.

Groman, Robert C. 1980. "Data base for Location-Dependent Data Detailed Design Specifications." Woods Hole Oceanographic Institution.

Groman, Robert C. 1974. "The Digital Data Library System: Library Storage and Retrieval of Digital Geophysical Data." Woods Hole Oceanographic Institution technical report, 74-68.

Hartman, G.K. 1978. The Information Explosion and Its Consequences for Data Acquisition, Documentation and Processing; An Additional Aspect of the Limits of Growth. World Data Center A for Solid Earth Geophysics; U.S. Department of Commerce, NOAA EDS, Boulder Colorado, Report UAG-65, also known as Report SE-11.

Heaps, H. S. 1978. Information Retrieval: Computational and Theoretical Aspects. New York: Academic Press.

Hilsenrath, Joseph and Breen, Bettijoyce. 1978. OMNIDATA An Interactive System for Data Retrieval, Statistical and Graphical Analysis, and Data-Base Management, A User's Manual. United States Department of Commerce, National Bureau of Standards Handbook 125.

Jenne, Roy L. and Joseph, Dennis H. 1974. "Techniques for the Processing, Storage and Exchange of Data." NCAR - TN/IA-93.

Kim, Won. 1979. Relational Database Systems. Computing Surveys. 11: 185-211.

Kou, Lawrence T. and Wong, C.K. 1975. A Note on the Set Basis Problem Related to the Compaction of Character Sets. CACM. 18: 656-657.

Knuth, Donald E. 1973. The Art of Computer Programming Volume 3, Sorting and Searching. Reading: Addison-Wesley.

Lipton, Richard J., Rosenberg, Arnold L. and Yao, Andrew C. 1980. External Hashing Schemes for Collections of Data Structures. JACM. 27: 81-95.

Little, James J. 1978. Strategies for Interfacing Geographic Information Systems, in Harvard Papers on Geographic Information Systems, Edited by Geoffrey Dutton, vol. 5, Data Structures: Surficial and Multi-Dimensional, Boston: Harvard University.

Martin, James. 1975. Computer Data-Base Organization, Englewood Cliffs: Prentice-Hall, Inc.

Molloy, A.M. 1977. Geographic Data Development. Mitre Corporation, Bedford, Mass.

Nagy, George and Wagle, Sharad 1979. Geographic Data Processing. Computing Surveys. 11: 139-181.

National Academy of Science. 1972. Understanding the Mid-Atlantic Ridge, A Comprehensive Program, The report of a special workshop under the auspices of the Ocean Science Committee of the National Academy of Science - National Research Committee Ocean Affairs Board, January 24-28.

National Academy of Science. 1976. Geophysical Data Centers, Impact of Data-Intensive Programs. Geophysical Data Panel, Committee on Data Interchange and Data Centers, Geophysics Research Board, Assembly of Mathematical and Physical Sciences, Washington D.C.

National Science Foundation. 1978. Grants for Scientific Research. Reference No. 78-41.

National Science Foundation 1979. Guide to Programs, Fiscal Year 1979. Stock #038-000-00386-1.

NOAA (National Oceanic and Atmospheric and Administration). 1977. Marine Geophysical Data Exchange Format - 'MGD77'. (Bathymetry, Magnetics, and Gravity). KGRD No. 10, Boulder: National Geophysical and Solar Terrestrial Data Center.

Nuclear Energy Agency. 1978. NEA Working Group on Nuclear Energy Information - OECD Nuclear Energy Agency, Paris.

Nunamaker, Jay F., Swenson, Donald E., and Whinston, Andrew B. 1973. GPLAN: A Generalized Data Base Planning System in Modeling and Simulation. Vol. 4 Pittsburgh: Instrument Society of America.

Peucker, T.K., Mark, D.M., and Pfaltz, J.L. 1973. Geographic Data Structures.

Proceedings of a Seminar on Variable Grid Resolution Issues and requirements. 1977. Hydrologic Engineering Center, Davis, California.

Severance, Dennis and Duhne, Ricardo. 1976. A Practitioner's Guide to Addressing Algorithms. CACM. 19: 314-326.

Smith, Stuart M. Scripps Institution of Oceanography, La Jolla, California. Interview, 1979.

Surkan, Alvin J. 1972. Organization of a National Resources Data Bank System. Spring Joint Computer Conference.

Wagle, Sharad G. 1978. "Issues in the Design of a Geographical Data Processing System: A Case-Study." Ph.D. Dissertation, University of Nebraska.

Weissel, Rose Anne. Lamont-Doherty Geological Observatory, Palisades, New York. Interview, 1979.

APPENDIX VI

Data Base for Location Dependent Data

Detailed Design Specifications

Last Modified: 13 January 1982

Robert C. Groman

Table of Contents

Table of Contents

(continued)

DICTIONARY STRUCTURE

Revision Date: 11 August 1981

The dictionary is maintained in two distinct parts of a direct access file. In the first part entries are inserted and retrieved via a straightforward hashing algorithm. Primary area overflow technique is used to resolve collisions. Each 48 byte record (except record 1) in this section is organized as follows:

Unique integer*2 term id number	34 byte word entry	Local integer*4 pointer to next record in alphabetical sequence	Local integer*4 pointer to first record of entry attributes (details pointer)	32 bits for flags. Bit 1 for synonym entry

Record 1 in the dictionary maintains the following information:

Next term id # available for use for word entry integer *2	First integer*4 pointer for alphabetically sequenced linked list	Modulo value for hashing scheme (equal to size of first part of dictionary) integer*4	Local integer*4 pointer into details section for system specifications and other info	Integer*2 of next number available as data group numbers	Next free record in second section of the dictionary integer*4

The direct access part of the dictionary containing the dictionary details contains the following kinds of records:

Record Type 1 - Library System parameters (subtypes 1, 2 and 3)

Record type = 1 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Record type subcode = 1 integer *2	Network name for data 14 bytes	Disk name for data 8 bytes	Directory name for data 14 bytes	Un-used 2 bytes

Record type =1 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Record type subcode =2 integer *2	Network name for bounds tables 14 bytes	Disk name for bounds tables 8 bytes	Directory name for bounds tables 14 bytes	Un- used 2 bytes

Record type =1 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Record type subcode =3 integer *2	Network name for ship table 14 bytes	Disk name for ship table 8 bytes	Directory name for ship table 14 bytes	Un- used 2 bytes

Record type 2 - Parameter location in data base

Record type 2 integer *2	Next local pointer for entry integer *4	Record type of next record in list integer *2	Owner pointer into first part of dictionary file integer*4	Data base file type containing this data parameter 1=primary 2=secondary integer *1	Data base group record number containing this data parameter integer *2

Starting byte number in data base integer *1	Length in bytes integer *1	Variable type code as stored in data base integer *2	Bit posi- tion starting from right integer *1	Original variable type code before conversion integer *2	Date added 5 bytes

Record type 3 - English definition (units specification follows definition)

Record type 3 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Subcode integer*2	definition of entry continued as needed 38 characters

Record type 5 - Units specification (follows definition; last entry)

Record type 5 integer *2	Next local pointer *4	Record type of next record in list integer *2	Subcode integer*2	Units of entry in text form (continued if needed) 38 characters

Record type 7 - Pointer variables

Record type 7 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Subcode integer*2	Reserved for pointer type variable 38 bytes

Record type 11   Relationship/set or table name variable

Record type 11 integer *2	Next local pointer integer *4	Record type of next record in list integer*2	Subcode integer*2	Reserved for relationship/ set/ table name/type variable 38 bytes

Record numbers between 23 and 43 specify an offset and range (low and high values) for each data type as appropriate.

Record type 23 - Integer *1, *2, or *4 type variable

Record type 23 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Mult. constant integer *1	Offset value integer *4	Lowest valid value (used as invalid value) integer *4	Highest valid value integer *4	Unused 27 bytes

Record type 29 - Real *4 type variables

Record type 29 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Mult. constant integer *1	Offset value Real *4	Lowest valid value (used as invalid value) Real*4	Highest valid value Real*4	Un-used 27 bytes

Record type 31 - Character type variable

Record type 31 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Not specified yet

Record type 37 - Bit data type

Record type 37 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Not specified yet

Record type 41 - Complex data type

Record type 41 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Not specified yet

Record type 43 - Double precision data type

Record type 43 integer *2	Next local pointer integer *4	Record type of next record in list integer *2	Not specified yet

In all cases, a value of 0 for a next pointer indicates that the linked list has terminated.

The name for the dictionary file is

DBLD.DIC

from Data Base for Location-Dependent data.

DATA STRUCTURE

Revision Date: 23 July 1981

The data are organized into three categories: primary data, secondary data, and pointers and file specification data.

Primary Data

The primary data include the date, time, and position information (see figure 1). These data are organized in direct access files according to the particular one degree geographical square in which the data resides. This approach is taken in a data base optimized for location dependent data as it provides:

1. growth potential for other than geophysical data types,
2. retaining the ability to reduce the on-line size of the data base while affecting as few users as possible,
3. allowing the data base to store data in an economical manner - without resorting to extraordinary pointer (find next record) schemes,
4. simplicity.

One implementation of this scheme could require as many as 64800 files to be supported if data were stored in all degree squares of the world. However, specific applications usually leave a significant fraction of the world's surface unsampled. In particular, the geophysical data base implemented here would need less than 50,000 position files even when all data are included in the data base. However, this is still too many files for the operating system to maintain, so an alternate approach is taken. Only four files are used to cover the world, one file for each quadrant of the earth (that is, north-east, north-west, south-east and south-west). Within each quadrant file, data can still be accessed by degree square because of the bounds table (see next section). This scheme retains the desired retrieval capabilities without undue file manipulation.

The file naming convention for primary data uses the format

(NORTH/SOUTH)(EAST/WEST).DB1

Each primary record is associated with (that is, points to) at least one pointer-type record, also contained in the position file. A pointer record contains a "next record" pointer for each relationship in which the position record participates. The pointer record contains a record number and a code for the file name. Since the location of the stored data depends on its geographical position it seems logical that this code depend on the latitude and longitude values.

In order to speed up searching and responses to queries on availability of data within specific degree squares, the number of points within a degree square for each cruise leg is maintained in the bounds table (see the following section). A pointer to the last record in the degree square for each cruise leg (included once whenever a cruise leg enters or reenters a degree square) permits faster navigating through the data base.

This approach offers a good deal of growth potential since a position record can participate in an essentially unlimited number of relationships without compromising the pointer structure. Also, it offers the opportunity to restructure the ordering of the pointer-type records (where there are more than one) to improve retrieval.

Since a sequential write operation is less costly than the direct access write we provide a means to add new data to these position files using the sequential write operation. This method also minimizes the file sizes by allowing files to be only as long as they need be, and without having to preallocate file space. The first record in each position file maintains information needed to maintain the link list structure of the file and supports this scheme. The format of the first record is as follows.

Record type =0 integer *2	First free record (equals record number of the last record in the file plus one) integer*4	Creation date year hour,day minutes 5 bytes	Date last updated year hour,day minute 5 bytes	

Secondary Data

The pointer record includes a pointer into a secondary data file. That is, any parameters (the secondary data) associated with the position are stored in a separate file. There are a number of advantages to this approach.

1. Adding a new data type does not affect the organization of the primary data and allows for a great deal of growth potential.

1. Secondary data can be removed entirely from the disk without affecting queries based on position.

3. Selected portions of secondary data can easily migrate on and off the disk as the needs for these data change.

4. The cell lengths in the primary and secondary files need not be the same, making better use of disk space.

The file naming convention for the secondary data is

(NORTH/SOUTH)(EAST/WEST).DB2

At the present time one attribute file name at most will exist for each positional file name. However, one direction of growth is to divide different attributes into different attribute file names.

The attribute (or secondary) file is maintained as a multi-list structure, with linked lists of attributes, one linked list for each parent (owner) position entry in the positional data file. Each linked list is made up of storage cells of disk space containing information for one group of data attributes and a next pointer. The purpose of the group concept is to improve retrieval times for data which are usually used together. Figure 2 contains the groupings for known data parameters. Note that the pointer record is located in the primary file rather than in the secondary file. This is done so that the secondary data files can be totally removed without affecting the ability of the query software to follow set/relationship next pointers throughout the data base.

The cell size for the attributes file should be at least 22 bytes long for the groups as presently defined. We chose a record size of 24 bytes.

Data Available Mask

The data available mask (defined in the primary record) contains information about which dictionary terms have data values stored with this position. The bit number within the mask corresponds to the term id number assigned to the term when it is added to the dictionary. Bit number 0 (zero) is reserved for future use. Bit number 1 must be used to define the POSITION(S) term. Unfortunately, this scheme is not as implementation dependent as the prime number scheme originally designed but it does allow faster access and more immediate growth potential.

Figure 1

Primary File Contents

Position/Time Record

Contents	Length in bytes	Starting byte
Record type (=1)	2	1
Next local pointer	4	3
Reserved_A	2	7
Year	1	9
Day and hour	2	10
Minute	2	12
Time zone	2	14
Latitude	4	16
Longitude	4	20
Z-coordinate or height (reserved)	4	24
Cumulative distance	4	28
Data available mask	8	32
Ship table owner pointer	4	40
Pointer to secondary data	4	44

Pointer Record (subcode = 1)

Contents	Length in bytes	Starting byte
Record type (=2)	2	1
Next local pointer	4	3
Reserved_A	2	7
Record subtype (=1)	2	9
Reserved_B	4	11
Next pointer in degree square	4	15
Reserved_C	2	19
Next file name code, cruise leg	4	21
Next pointer for cruise leg	4	25
Reserved_D	2	29
Next file name code, abbreviated navigation stream	4	31
Next pointer for abbreviated navigation stream	4	35
Reserved_E	2	39
Reserved_F	8	41

Figure 1 (continued)

Counter record (subcode = n)

Contents	Length in bytes	Starting byte
Record type (=3)	2	1
Next local pointer	4	3
Reserved_A	2	7
Record subcode = n, defined by variable id # from the dictionary	2	9
Number of points in degree square in this cruise leg	4	11
Backwards pointer to ship table record type 2	4	15
Number of times queried	4	19
Date last queried	5	23
Date last updated	5	28
File code for first position pointer	4	33
Record number for first position	4	37
File code for last position pointer	4	41
Record number for last position	4	45

Figure 2

Secondary File Contents

Navigation detail	Length in bytes	Starting byte
record type (= 3)	2	1
next local pointer	4	3
reserved	2	7
current speed east-west	2	9
current speed north-south	2	11
navigation quality code	1	13
navigation type code	1	14
platform (ship) speed east-west	2	15
platform (ship) speed north-south	2	17

Bathymetry detail	Length in bytes	Starting byte
record type (= 5)	2	1
next local pointer	4	3
reserved	2	7
depth correction code	1	9
depth quality code	1	10
depth type code	1	11
uncorrected depth (travel time)	4	12

Magnetics detail	Length in bytes	Starting byte
record type (= 7)	2	1
next local pointer	4	3
reserved	2	7
diurnal correction	2	9
magnetic quality code	1	11
magnetic sensor altitude	2	12
sensor used code	1	14
geomagnetic total field, 1st sensor	4	15
geomagnetic total field, 2nd sensor	4	19

Figure 2 (continued)

Gravity detail	Length in bytes	Starting byte
record type (= 11)	2	1
next local pointer	4	3
reserved	2	7
Bouguer anomaly	2	9
Eotvos correction	2	11
gravity quality code	1	13
observed gravity	4	14

Seismic detail	Length in bytes	Starting byte
record type (= 13)	2	1
next local pointer	4	3
reserved_A	2	7
seismic shot identification	12	9
reserved_B	4	21

Geophysical detail	Length in bytes	Starting byte
record type (= 2)	2	1
next local pointer	4	3
reserved_A	2	7
corrected depth	4	9
geomagnetic anomaly	2	13
free air anomaly	2	15
reserved_B	8	17

Bounds Table

One of the basic features of any data base management system is that an entity may participate in one or more relationships (or sets) with other entities. In the data base optimized for location-dependent data it follows that geographical proximity is one such relationship. The organization chosen to implement this relationship is a two level structure, called the bounds table.

The first level in the bounds table data structure consists of 648 file names corresponding to the 648 different 10 degree by 10 degree geographical shapes on the world's surface. (We should not call these shapes squares because of the spherical shape of the earth). Each file name is defined by the particular 10 degree by 10 degree area it refers to, as

BND (N/S) p (E/S) qq.TBL

where N means north, S means south, E means east, and W means west; p can take the values 0 through 9; and qq can take the values between 00 and 18. Leading zeros must be present. This file naming convention makes it easier to identify which 10 degree area a table refers to.

In order to save disk storage, if no data exist in a 10 by 10 degree area then the corresponding bounds table as defined above will not exist.

The second level in the bounds table data structure contains the contents of these files. Each file contains an entry for each one degree shape ("square") within the 10 degree shape. The formula for placement or retrieval of a specific 1 degree entry is

$$\text{linear position} = \text{ABS(latitude units)} \times 10 + \text{ABS(longitude units)} + 2$$

This formula causes a progression from left to right (west to east) and then bottom to top (south to north) for positive latitudes and longitudes. There will always be at least 101 entries in the file although many entries may indicate that no data exists in that one degree shape.

As an example, consider the 10 degree shape whose upper right corner is at 10 degrees south and 150 degrees west. The corresponding bounds table is called

BNDS1W15.TBL

The data located within the boundaries equal to or less than 10 degrees south but greater than 20 degrees south, and equal to or less than 150 degrees west but greater than 160 degrees west would be accessible through this file. In this case the table name refers to the upper right corner because we have a negative latitude and longitude.

The contents of an entry in the bounds table include the number of points within the one degree shape. A local pointer is present which points to additional information about this degree square. The additional information is maintained in a linked list structure. The records in this structure maintain counts and pointers for each cruise leg as it enters and leaves the degree shape. Records for each data type indicate the total number of points available in the degree shape for each data type. Records containing additional information are kept in this same file, after record number 101. The formats for the various 48 byte record types follow:

Record Number 1:

Record type =0 integer*2	Next free record integer*4	Creation date 5 bytes	Date last updated 5 bytes	

Record type 2: for records 2 through 101, count of the total number of navigational records in the degree square

Record type (=2)	2	1
Next local pointer	4	3
Next record type	2	7
Record subcode = 1,	2	9
Total number of navigation points in the degree shape	4	11
Reserved_A	4	15
Number of times queried	4	19
Date created	5	23
Date last updated	5	28
Reserved_B	4	33
Reserved_C	4	37
Reserved_D	4	41
Reserved_E	4	45

Record numbers greater than 101 contain information about each cruise leg as the ship's track enters and leaves a degree square (record type 3) as well as a totals count for each data type available in the degree square (record type 2, with subcode greater than 1).

Record type 2: totals count for the data type code specified by the subcode.

Record type (=2)	2	1
Next local pointer	4	3
Next record type	2	7
Record subcode = n,	2	9
Total number of data points in the degree shape	4	11
Reserved_A	4	15
Number of times queried	4	19
Date created	5	23
Date last updated	5	28
Maximum value (*)	4	33
Minimum value (*)	4	37
Reserved_D	4	41
Reserved_E	4	45

Record type 3: pointer, counter and statistics for cruise leg navigation records each time a cruise enters this degree square.

Record type (=3)	2	1
Next local pointer	4	3
Next record type	2	7
Record subcode = 1,	2	9
Number of points in degree square in this cruise leg segment	4	11
Backwards pointer to ship table record type 2	4	15
Number of times queried	4	19
Date last queried	5	23
Date last updated	5	28
File code for first position pointer	4	33
Record number for first position	4	37
File code for last position pointer	4	41
Record number for last position	4	45

(*) Values are stored as real*4, in the original units. Complex values have only their real values stored. Character string variables cause a value of 0 (zero) to be stored.

This two level scheme allows for the fact that there will be some 10 degree shapes containing no data. Therefore, less than the full number of 648 file names will be needed. With this scheme there is little overhead due to sparse data sets.

Any data residing on the International Date Line is taken to be at 180 degrees west rather than 180 degrees east. A latitude or longitude of exactly zero is treated as a positive number.

Ship Table

Another relationship that geophysical measurements participate in is the commonality of collection platform. Typically, this platform is an ocean-going ship. While different organizations have chosen unique ways to identify data collected by one ship during one collection period, a common approach is to call the collection period from one port stop to another a leg. It is also common to group these legs together into a cruise, where a cruise begins and ends at the home port. Each leg is numbered sequentially with the first leg, leg 1, departing from the home port.

Since it is very common for researchers to request data from one or more specific cruise legs, the data base scheme uses a ship table (similar to the bounds table) to facilitate such request retrievals. Besides holding a pointer (file name and record number) to the first position record, the ship table holds important information about the cruise including the unique cruise name, start and end dates, chief scientist name during the leg, number of data points collected, project name, type of instrumentation used, types of data collected, etc.

The bounds table (described above) and the ship table could be intertwined by identifying whenever a cruise leg enters and leaves a degree shape. In such a scheme the ship table would contain a pointer to a linked list of pointers. These pointers would point to the first position inside a new degree shape, following the cruise track. The bounds table would contain a pointer to a similar linked list but in this case all the positions referred to would lie in the same degree shape. Advantages of this scheme include: a simple way of identifying when data within a particular degree shape belong to a particular cruise leg; and an easy method to respond to queries about which cruises and how many data points are available in specific areas. A disadvantage is that simplicity is reduced by forever tying together two independent relationships. Adding new relationships could overly complicate the data base structure. Because of these disadvantages this scheme is not used.

Insertion and retrieval from the ship table is via a simple hashing algorithm similar to the one used in the dictionary structure. Based on current data holdings and estimated insertion rates, the ship table need be only 307 entries long, resulting in a hash table less than 60 percent full. However, like the dictionary file, the ship table has a primary and secondary part. The contents of the primary part are identified as record type 1. The secondary part contains further information about the cruise.

The formats for the various 48 byte record types in the ship table file called SHIP.TBL follow.

Record number 1:

Reserved 2 bytes	First pointer to alpha-betical list integer*4	Modulo value for hashing integer*4	Local pointer for details integer*4	Reserved 2 bytes	Next free record in second section integer*4

Record type 1: Ship name

Reserved 2 bytes *2	Ship identifi-cation 34 bytes	Next pointer in alpha-betically sorted linked list integer *4	Details pointer integer*4	Flags Bit 1 for synonym 2 bytes (16 bits)	Security code integer*2

Note that the pointer to the first data point in the cruise does not appear here. It will appear in a linked list structure defined in record type 2 of the ship table.

Record type 2 - cruise start and end dates, date ship table entry was made, and backwards pointer to ship id.

Record type =2 integer *2	Next record in linked list integer *4	Record type of next record in list integer *2	Date the entry is made in the table 5 bytes	Start date of cruise leg year hour/day minute (5 bytes)	End date of cruise leg year hour/day minute (5 bytes)

Backwards pointer to ship name integer*4

The backwards pointer is made available so that the first part of the ship table can change in size without having to change each position. (Each position record contains a backwards pointer to its corresponding ship identification.) But in order for this to be useful, the ship table detail section must grow from the end of the preallocated file backward towards record one.

Record type 3 - pointer, counter and statistics for data parameters, identified by the subcode value n

Record type (=3)	2	1
Next local pointer	4	3
Next record type	2	7
Record subcode = n,	2	9
Number of points of this data type in cruise leg	4	11
Backwards pointer to ship table record type 2	4	15
Number of times queried	4	19
Date last queried	5	23
Date last updated	5	28
File code for first position pointer	4	33
Record number for first position	4	37
File code for last position pointer	4	41
Record number for last position	4	45

As soon as any data are added to the data base, there must exist a record type 3 for the position information. The record subtype (or subcode) for the position information is 1. There exists a record type 3 for each data type available in the cruise leg. The subcodes for these data types are the same as their unique dictionary identifying numbers. These records are usually placed immediately after the record type 2.

Figure 3 summarizes the secondary ship table types 5 through 19. All have the same format as the example given in the figure.

Figure 3

Secondary Ship Table Parameters

Description	Record type	Subcode
Port stops	5	0
Chief scientist	7	0
Contributor	11	0
Project name	13	0
Comments	17	0
MGD77 header	19	1 thru 24

Record type 5 integer *2	Next local pointer *4	Record type of next record in list integer *2	Subcode integer*2	Port stops (continued if needed) 38 characters

Record type 23 is used to store the overall latitude and longitude bounds for the cruise leg. The values are stored in compact format. The contents of record type 23 follow:

Record type 23 - overall latitude and longitude bounds for the cruise leg

Record type (=23)	2	1
Next local pointer	4	3
Next record type	2	7
Reserved_A	2	9
Maximum latitude	4	17
Minimum latitude	4	21
Maximum longitude	4	25
Minimum longitude	4	29
Reserved_B	4	33
Reserved_C	4	37
Reserved_D	4	41
Reserved_D	4	45

USER DEFAULT FILE

6 June 1979

The user default file contains attributes about each user of the DBLD. These attributes are used to define the default options and conditions for various segments of the retrieval software. The default file name is contained in the validation file.

UNIT REFERENCE NUMBER ASSIGNMENTS

Revision Date: 6 April 1980

A number of unit reference numbers are permanently reserved. These are

validation file -- 20
dictionary file -- 21
history/accounting file -- 22
help file -- 23
ship table file -- 24
bounds table file -- 25
reserved -- 26 -- 39
(for the tables)

Unit reference numbers from 40 through 49, and 50 through 59 are available for assigning to various data files, corresponding to one degree squares of data. The numbers in the forties are to be used for primary data files; the numbers in the fifties are to be used for the secondary data files. Unit reference numbers 10 through 19 are used in short term situations, usually within one software module.

HISTORY AND ACCOUNTING STATISTICS

Revision Date: 14 April 1980

Each time a data base user begins a DBLD procedure an initial entry is made in the history/accounting file. Additional entries (see Appendix 3) are made at various times indicating:

data base modules used
which data are accessed
number of points read
.
.
.

At the end of the program a final entry is made in the history/accounting file. Using the information stored from the first and last entries, it is possible to compute the number of buffered input/output operations, CPU time, direct input/output operations, elapsed time, and page faults.

These entries do not reflect the 'effort' required to bring the program to the point where it writes the first entry into the history/accounting file. For example, in the case of program UPDBDIC (UPdate the Data Base DICtionary), in a test run made 19 October 1979, the following differences were noted:

	Computed	TIMRB/TIMRE
elapsed time(sec.)	547.63	555.94
cpu time (sec.)	2.13	2.31
direct I/O	57	59
buffered I/O	281	294
page faults	192	256

The above figures indicate that the history/accounting file is adequate for our purposes.

These statistics can be used to compute a charge for the services performed by the DBLD software as well ascertain use patterns. They can be further studied to help reorganize and modify the data stored in the data base.

RETRIEVAL COMMAND FORMAT

Revision Date: 26 October 1979

The general format for a retrieval command is

```
     !-         -!   !-  !-        -! -!
 verb!/switch   !... !   !object    !, !...
     !_         _!   !_  !_        _! _!
```

where 'verb' is an acceptable retrieval command action verb such as DISPLAY, DEFINE, HELP, LIST, SHOW etc.; '/switch' is zero, one or more verb modifiers separated by a slash; and 'object' is zero, one or more parameters separated by a comma.

For example, to obtain the definition of the dictionary terms LONGITUDE and YEAR enter the command

DEFINE LONGITUDE,YEAR

In this case, DEFINE is the verb and LONGITUDE and YEAR are objects. There are no modifiers. Since unique abbreviations are permitted for verbs and switches this command could be written as

DEF LONGITUDE,YEAR

In general, objects cannot be abbreviated.

Consult the help files for a current list of commands and options.

Appendix 1

Revision Date: 4 Jan 1982

Definition of Status Codes

Code	Meaning
0	no error, operation successful
1	some error has occurred
4	two dates are not the same
6	file open error
7	direct read error
8	direct write error
9	end of file condition
10	record not found in a linked list
11	file does not exist
12	array index out of range in MAXMINVAL
13	No valid data values contained in this detail record. That is, all data values are the null value.
14	write error to output file
15	incorrect record type
16	unknown or not implemented record type
17	more than 64 occurrences of record type 3 in ship table for current cruise-leg.
18	COPY operation cancelled by user
19	COPY operation cancelled because no cruises specified and bounds too large (more than 100 10-degree squares).
20	error in GETNXTPNO - existing file could not be opened.
21	option not available (PRSSWTCH)
22	abbreviation not unique (PRSSWTCH)
99	end of library query procedure

Appendix 2

Revision Date: 5 Oct 1981

Counter/Statistic Assignment Numbers

Meaning	Counter Number
data base primary records output	1
data base detail parameters	numbers as assigned by the dictionary
number of navigation records input	130
number of times primary files opened for append	131
number of times detail files opened for append	132
number of times bounds tables opened	133
number of times detail files opened for read only	134
number of times primary files opened for read only	135
number of times primary files opened for read and write	136
number of times detail files opened for read and write	137
number of primary files created and opened for sequential appending	138
number of detail files created and opened for sequential appending	139

Appendix 3

Revision Date: 24 Aug 1981

Formats for Messages to History File

Routine Name	Contents
ADDNEWUSR	'Authorized new user', followed by the user name.
DBOPNUPD	'***START STATISTICS***'
RDSTBLCTS	ship owner pointer to record type 2, start location for reading and subcode value equal to the data parameter identification number, in the format (I12,2X,I12,2X,I6)

Appendix 4

Revision Date: 23 July 1981

Setting File Access Protection

On the VAX-11/780 computer, computing access to user files is controlled by setting the protection levels for each file. Certain files in this library need to be accessed (read only) by all users while other files may need to allow writing to them by all users. The following summary describes the procedures to follow in order to set the correct access authorization for files on the VAX computer.

The account under which the library or dictionary updates are made should have as the default authorization:

```
system - read, write, execute, delete
owner  - read, write, execute, delete
group  - read, execute
world  - read, execute
```

Using these protection levels will allow everyone to have read access to the data and library files and to have execute privileges for the programs. Note that all program sources should be made secure by allowing only the owner to have read access to the source files.

These default authorization parameters are appropriate for all except certain files. The authorization for the history/accounting file must be set to the following:

```
system - read, write
owner  - read, write, delete
group  - read, write
world  - read, write
```

To change the authorization parameters on the history/accounting file issue the following VAX command:

```
$SET PROTECTION=(SYSTEM:RW,OWNER:RWED,GROUP:RW,WORLD:RW)
                BIGA:[DBDICT]DBHISTORY.ACC
```

Appendix 5

Revision Date: 6 June 1980

Validation File

The validation file contains a list of authorized users of the library system. There exist three levels of authorization as follows:

Level #	Authority
1	all actions permitted
2	all actions permitted except authorizing new users to the libary. Hence, this user can add new date, delete and modify data.
3	read only access to the library is permitted this user level. This is considered the proper authorization for a user expected to do typical data retrieval and queries.

The format for an entry in the validation file follows:

user name	12 characters, left justified
data authorized	16 characters
validation level	integer*2 (encoded as I4)
security level	integer*2 (encoded as I2)
user default file	40 character file specification

The security level is used to control access to specific cruise legs of data. Access is permitted to any cruise leg whose security code (as stored in the ship table) is equal to or lower than the user's security level value.

Whenever a procedure is initiated (i.e. a program is 'run') a check is made against the validation file. The user at his or her terminal (or in batch) must be authorized at the appropriate validation level in order to continue the chosen task.

Appendix 6

Revision Date: 24 August 1981

System Manager's Guide

System Start-up Procedures

Run program INITDBLD creating the dictionary, history, and validation files as necessary. Be sure to follow the file access instructions in Appendix 4.

Sample run of INITDBLD:

$ RUN INITDBLD.EXE	
Y	(Yes, create a new dictionary)
0	(Modulo value, 0 is default value)
::	(Network name for data values)
BIGA:	(Disk name for data values)
[DBDATA]	(Directory name for data values)
::	(Network name for bounds table)
BIGA:	(Disk name for bounds table)
[DBTBLE]	(Directory name for bounds table)
::	(Network name for ship table)
BIGA:	(Disk name for ship table)
[DBTBLE]	(Directory name for ship table)
Y	(Yes, create a new history/accounting file)
Y	(Yes, create a new validation file)

Y	(Yes, create a new ship table)
0	(Modulo value for the ship table hashing function. 0 is default value)

Run program UPDBDIC. This program updates the dictionary structure with the data attributes which will be stored in the library. The first entry must be the term POSITION(S) as follows:

POSITION(S)	(new term)
N	(not a synonym)
1	(group #)
1	(primary file type)
14	(starting byte #)
12	(length)
INTEGER	
INTEGER	

the latitude, longitude and height (z coordinate) above or below sea level identifying the geographical location on, over or below the earth's surface

degrees for latitude and longitude, meters for height

0	(multiplication factor)
0	(offset)
0	(lowest value)
0	(highest value)

End the input with a Control Z.

You should note and record what group numbers are assigned to each secondary data attribute so that you can generate the correct read/write routine for each record type (identified by group number). Since the software does not automatically control how data are stored, you must decide how to store the secondary data attributes. Then, store the data attribute names in an order that will cause the group number to be assigned correctly. Whenever you enter a 0 (zero) for the group number, the system automatically assigns the next available group number to the attribute being entered.

Periodic Maintenance

The history file, DBHISTORY.ACC, as defined during the system start-up, should be listed and removed from disk onto magnetic tape whenever it becomes large or whenever it is time to do the charge-back operation. You must create a new history/accounting file using the INITDBLD program and change the protection for this file as outlined in Appendix 4.

Review the history files to see what secondary data might be removed from the system without affecting too many users. This procedure will minimize the cost of maintaining an on-line data library.

Review the status of the bounds tables after each program DBINSERT run to insure that sufficient space is available for more new data. Also check the ship table for available space. Both primary and secondary files will grow as needed.

Authorizing New System Users

New users are authorized for data insert and/or retrieval by using program DBVALID. The program requests the the following input:

User name	enter the VAX user name as assigned by the VAX operations staff.
Validation level #	see appendix 5 for a description of the validation levels
Security level #	see appendix 5 for a description of the security levels.
User default file name	this file name, complete with disk, directory and file name, is executed via the @ command by DBQUERY whenever this user begins execution of the DBQUERY program. The file must exist even if it contains no commands.

When the user validation file is first created, by running program INITDBLD, the username of the process executing this program is granted system manager status. This user is authorized to add new users to the system. Additional people can be granted system manager status by assigning the appropriate authorization level number. The file MANAGER.DBQ is set up to be the command file executed when DBQUERY is first executed for this user.

Appendix 7

Revision Date: 13 January 1982.

Program Version Number and Date

Program Name	Version Number	Date
DBINSERT	1.10	8 January 1981
DBQUERY	1.10	11 July 1982
DBVALID	1.00	6 June 1980
SHPTBLINS	1.10	11 January 1982
UPDBDIC	1.00	5 October 1979

Appendix 8

Revision Date: 18 September 1981

Steps to Add New MGD 77 Data

1. Insure the cruise leg identifier has been added to the shiptable via the SHPTBLINS program.

2. Run program CHECKMGD to check for time and navigation errors in MGD input format.

3. Create a batch job for DBINSERT under the GROMAN account. Unit 1 is assigned to the input data in MGD 77 format as follows:

```
$ ASSIGN A2093L01.MCD  FOR001
$ RUN DBINSERT
A2093L01
```

4. Submit the batch job to run at night, during class 3. One leg takes about 20 minutes of CPU time.

5. After successful DBINSERT, do:

   a. Backup of data base files (DBSAVE.COM)

   b. Backup of DBINSERT.BAT and MGD data files (MGDSAVE.COM)

   c. Delete MGD data files from disk.

6. If DBINSERT job was not successful, restore data base files from most recent saveset using DBRESTORE.COM command procedure.

MANDATORY DISTRIBUTION LIST

FOR UNCLASSIFIED TECHNICAL REPORTS, REPRINTS, AND FINAL REPORTS
PUBLISHED BY OCEANOGRAPHIC CONTRACTORS
OF THE OCEAN SCIENCE AND TECHNOLOGY DIVISION
OF THE OFFICE OF NAVAL RESEARCH

(REVISED NOVEMBER 1978)

1 Deputy Under Secretary of Defense
(Research and Advanced Technology)
Military Assistant for Environmental Science
Room 3D129
Washington, D.C. 20301

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217
3 ATTN: Code 483
1 ATTN: Code 460
2 ATTN: 102B

1 CDR Joe Spigai, (USN)
ONR Representative
Woods Hole Oceanographic Inst.
Woods Hole, MA 02543

Commanding Officer
Naval Research Laboratory
Washington, D.C. 20375
6 ATTN: Library, Code 2627

12 Defense Technical Information Center
Cameron Station
Alexandria, VA 22314
ATTN: DCA

Commander
Naval Oceanographic Office
NSTL Station
Bay St. Louis, MS 39522
1 ATTN: Code 8100
1 ATTN: Code 6000
1 ATTN: Code 3300

1 NODC/NOAA
Code D781
Wisconsin Avenue, N.W.
Washington, D.C. 20235

1 Mr. Michael H. Kelly
Administrative Contracting Officer
Department of the Navy
Office of Naval Research
Eastern/Central Regional Office
Building 114, Section D
666 Summer Street
Boston, MA 02210

Woods Hole Oceanographic Institution
WHOI-82-27

COMPUTER STORAGE AND RETRIEVAL OF POSITION-DEPENDENT DATA by Robert Carl Groman. 139 pages. June 1982. Prepared for the Office of Naval Research under Contracts N00014-79-C-0071; NR 083-004 and N00014-82-C-0019; NR 083-004 and JOI, Inc.

A data storage and retrieval scheme has been designed and implemented which provides cost effective and easy access to location-dependent, 'geophysical' data. The system is operational on a Digital Equipment Corporation VAX II/780 computer. Values of measured and computed geophysical parameters, such as geomagnetic field, water depth and gravity field, are stored in the library system. In addition, information about the data, such as port stops, project name and funding agency are also saved. These data are available to a time sharing user, validated to use the software package, through a query language designed to interact with this data library. The data can be searched and retrieved both sequentially and geographically.

1. Data retrieval
2. Location-dependent data
3. Geographical retrieval
I. Groman, Robert Carl
II. N00014-79-C-0071; NR 083-004
III. N00014-82-C-0019; NR 083-004

Woods Hole Oceanographic Institution
WHOI-82-27

COMPUTER STORAGE AND RETRIEVAL OF POSITION-DEPENDENT DATA by Robert Carl Groman. 139 pages. June 1982. Prepared for the Office of Naval Research under Contracts N00014-79-C-0071; NR 083-004 and N00014-82-C-0019; NR 083-004 and JOI, Inc.

A data storage and retrieval scheme has been designed and implemented which provides cost effective and easy access to location-dependent, 'geophysical' data. The system is operational on a Digital Equipment Corporation VAX II/780 computer. Values of measured and computed geophysical parameters, such as geomagnetic field, water depth and gravity field, are stored in the library system. In addition, information about the data, such as port stops, project name and funding agency are also saved. These data are available to a time sharing user, validated to use the software package, through a query language designed to interact with this data library. The data can be searched and retrieved both sequentially and geographically.

1. Data retrieval
2. Location-dependent data
3. Geographical retrieval
I. Groman, Robert Carl
II. N00014-79-C-0071; NR 083-004
III. N00014-82-C-0019; NR 083-004

Woods Hole Oceanographic Institution
WHOI-82-27

COMPUTER STORAGE AND RETRIEVAL OF POSITION-DEPENDENT DATA by Robert Carl Groman. 139 pages. June 1982. Prepared for the Office of Naval Research under Contracts N00014-79-C-0071; NR 083-004 and N00014-82-C-0019; NR 083-004 and JOI, Inc.

A data storage and retrieval scheme has been designed and implemented which provides cost effective and easy access to location-dependent, 'geophysical' data. The system is operational on a Digital Equipment Corporation VAX II/780 computer. Values of measured and computed geophysical parameters, such as geomagnetic field, water depth and gravity field, are stored in the library system. In addition, information about the data, such as port stops, project name and funding agency are also saved. These data are available to a time sharing user, validated to use the software package, through a query language designed to interact with this data library. The data can be searched and retrieved both sequentially and geographically.

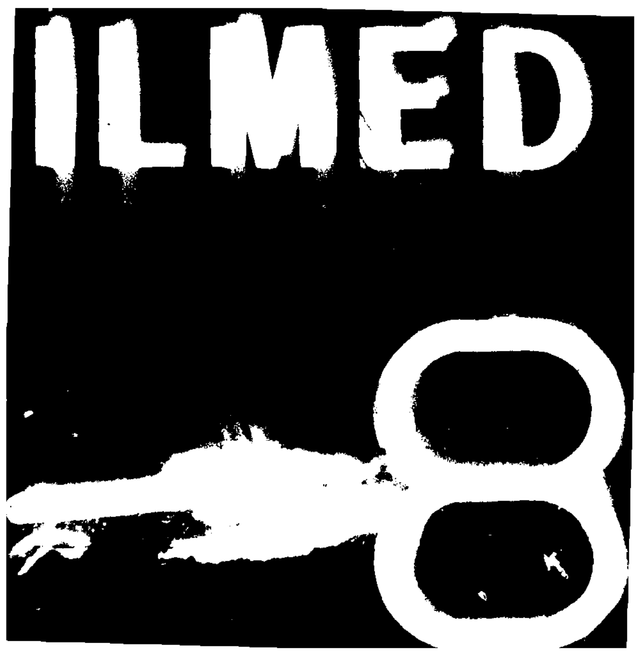
1. Data retrieval
2. Location-dependent data
3. Geographical retrieval
I. Groman, Robert Carl
II. N00014-79-C-0071; NR 083-004
III. N00014-82-C-0019; NR 083-004

Woods Hole Oceanographic Institution
WHOI-82-27

COMPUTER STORAGE AND RETRIEVAL OF POSITION-DEPENDENT DATA by Robert Carl Groman. 139 pages. June 1982. Prepared for the Office of Naval Research under Contracts N00014-79-C-0071; NR 083-004 and N00014-82-C-0019; NR 083-004 and JOI, Inc.

A data storage and retrieval scheme has been designed and implemented which provides cost effective and easy access to location-dependent, 'geophysical' data. The system is operational on a Digital Equipment Corporation VAX II/780 computer. Values of measured and computed geophysical parameters, such as geomagnetic field, water depth and gravity field, are stored in the library system. In addition, information about the data, such as port stops, project name and funding agency are also saved. These data are available to a time sharing user, validated to use the software package, through a query language designed to interact with this data library. The data can be searched and retrieved both sequentially and geographically.

1. Data retrieval
2. Location-dependent data
3. Geographical retrieval
I. Groman, Robert Carl
II. N00014-79-C-0071; NR 083-004
III. N00014-82-C-0019; NR 083-004


ILMED